

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

До захисту допущено:

Завідувач кафедри

_____ Олександр Коваль

«__» _____ 2020 р.

Дипломна робота

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Інформаційні технології моніторингу
довкілля»**

спеціальності 122 «Комп'ютерні науки та інформаційні технології»

**на тему: «Розробка веб-додатку для вивчення англійської мови у вищому
навчальному закладі»**

Виконав:

студент IV курсу, групи ТМ-61

Якимів Іван Андрійович _____

Керівник:

Доцент, к.т.н.

Ходаківський Олексій Володимирович _____

Рецензент:

Старший Викладач, к.т.н.

Баранюк Олександр Володимирович _____

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів без
відповідних посилань.

Студент (-ка) _____

Київ – 2020 року

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти перший рівень

Напрямок підготовки 122 Комп'ютерні науки та інформаційні технології

Спеціалізація Інформаційні технології моніторингу довкілля

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр Коваль
(підпис)

” ____ ” _____ 2020р.

ЗАВДАННЯ

на дипломну роботу студенту

Якимів Іван Андрійович

(прізвище, ім'я, по батькові)

1. Тема роботи Розробка веб-додатку для вивчення англійської мови у вищому навчальному закладі

Керівник роботи _____ доцент, к.т.н. Ходаківський Олексій Володимирович
(прізвище, ім'я, по батькові науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від “25” травня 2020р. № **1168-с**

2. Строк подання студентом роботи 10 червня 2020 року

3. Вихідні дані до роботи веб-додаток реалізований з допомогою мов програмування PHP, JavaScript, мовою розмітки HTML, мовою опису зовнішнього вигляду документа CSS, та формальної непроцедурної мови програмування MySQL. Для запуску програмного модуля необхідно встановити OpenServer.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) виконати аналіз існуючих систем вивчення англійської мови, розробити базу даних проекту, розробити програмну частину веб-додатку, здійснити програмну реалізацію спроектованого веб-додатку вивчення англійської мови, провести тестування і налагодження застосунку.

5. Перелік ілюстративного матеріалу титульний аркуш, постановка задачі, існуючі системи дистанційного навчання, архітектура розроблюваної системи, функціонал клієнтського додатку, висновки.

6. Дата видачі завдання ” 11 ” жовтня 2019 р.

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів виконання дипломної роботи | Термін виконання етапів роботи | Примітки |
|-------|---|--------------------------------|----------|
| 1. | Затвердження теми роботи | 12.12.19 | |
| 2. | Вивчення та аналіз задачі | 25.01.20 | |
| 3. | Розробка архітектури та загальної структури системи | 08.02.20 | |
| 4. | Розробка структур окремих підсистем | 12.02.20 | |
| 5. | Програмна реалізація системи | 26.02.20 | |
| 6. | Оформлення пояснювальної записки | 07.06.20 | |
| 7. | Захист програмного продукту | 18.05.20 | |
| 8. | Передзахист | 08.06.20 | |
| 9. | Захист | 16.06.20 | |

Студент _____ Якимів І. А. _____
(підпис) (прізвище та ініціали,)

Керівник роботи _____ Ходаківський О. В. _____
(підпис) (прізвище та ініціали,)

АНОТАЦІЯ

Метою цієї роботи є створення веб-додатку для зручного та продуктивного вивчення англійської мови у вищих навчальних закладах.

Веб-додаток реалізований з допомогою мов програмування PHP, JavaScript, мовою розмітки HTML, мовою опису зовнішнього вигляду документа CSS та формальної непроцедурної мови програмування MySQL. Для запуску програмного модуля необхідно встановити OpenServer.

Загальний обсяг роботи: 56 сторінок, 30 ілюстрацій та 10 бібліографічних найменувань.

Ключові слова: веб-додаток, англійська мова, інформаційні технології, база даних, програмний продукт.

ABSTRACT

The purpose of this work is to create a web application for convenient and productive learning of English in higher education.

The web application is implemented using PHP programming languages, JavaScript, HTML markup language, CSS document description language and the formal non-procedural MySQL programming language. To run the software module, you must install OpenServer.

Total volume of work: 56 pages, 30 illustrations and 10 bibliographic titles.

Keywords: web application, English language, information technologies, database, software product.

ЗМІСТ

| | |
|--|----|
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ..... | 7 |
| ВСТУП..... | 8 |
| 1 ПОСТАНОВКА ЗАДАЧІ..... | 10 |
| 2 АНАЛІЗ ПРОБЛЕМИ ВИВЧЕННЯ АНГЛІЙСЬКОЇ МОВИ | 12 |
| 2.1 Загальні відомості | 12 |
| 2.2 Роль англійської мови у сучасному світі..... | 13 |
| 2.3 Роль англійської мови у житті програміста..... | 15 |
| 2.4 Способи вивчення англійської мови | 16 |
| 2.5 Моделі додатків для вивчення англійської мови | 20 |
| 2.6 Виділення основних проблем | 22 |
| 2.7 Визначення основних вимог до ПП | 23 |
| 2.8 Висновки до розділу 2 | 30 |
| 3 ЗАСОБИ РОЗРОБКИ | 31 |
| 3.1 PHP - мова сценаріїв | 31 |
| 3.2 JavaScript - об'єктно-орієнтована мова програмування | 34 |
| 3.3 HTML - мова розмітки гіпертексту | 37 |
| 3.4 CSS - каскадні таблиці стилів | 38 |
| 3.5 MYSQL - мова структурованих запитів | 40 |
| 3.6 Висновки до розділу 3 | 43 |
| 4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ | 44 |
| 4.1 Формування бази даних..... | 44 |
| 4.2 AJAX – JS технологія..... | 49 |
| 4.3 jQuery – бібліотека JS | 50 |
| 4.4 Запити до бази даних MySQL за допомогою PHP..... | 53 |
| 4.5 Висновки до розділу 4 | 54 |
| 5 РОБОТА КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ | 55 |
| 5.2 Системні вимоги | 55 |

| | |
|---|----|
| 5.2 Сценарій роботи користувача від імені студента | 56 |
| 5.3 Сценарій роботи користувача від імені викладача | 63 |
| 5.4 Висновки до розділу 5 | 63 |
| ВИСНОВКИ | 64 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ | 65 |
| ДОДАТОК 1 | 66 |
| ДОДАТОК 2 | 68 |
| ДОДАТОК 3 | 79 |

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД – Бази Даних

ПП – Програмний Продукт

ТЗ – Технічне Завдання

IT – Information Technology

PHP – Hypertext Preprocessor

СКБД – Система Керування Базами Даних

MySQL – Structured Query Language

JS –JavaScript

DOM – Document Object Model

CSS – Cascading Style Sheets

HTML – HyperText Markup Language

ВСТУП

Не буде перебільшенням сказати, що англійська мова є найважливішою мовою у світі. Майже половина населення світу говорить англійською. Деякі найбільші країни у світі є англomовними, в тому числі США, Канада, Австралія і Великобританія.

Англійською мовою говорять більше ніж 1 мільярд людей на всій земній кулі, вона є рідною мовою 12 націй і офіційною мовою більш ніж 33 національностей. В якості другої мови, англійською говорять більш ніж 600-650 мільйонів громадян в 60 країнах. Це означає, що кожна сьома людина у світі говорить англійською мовою. Вона використовується урядом в міжнародних переговорах та підписаннях законів. Підприємці всього світу використовують її для торгівлі та підписання контрактів. Транспортні системи Землі підтримують комунікацію на морі, в повітрі та на суші, саме англійською мовою. Та що там говорити навіть світові змагання, такі як Олімпійські ігри чи Євро вибрати офіційною мовою саме англійську.

Англійську мову викладають у всьому світі. Завдяки знанням цієї мови можна поїхати навчатись у найпрестижніші університети світу. Та навіть за відсутності змоги подорожувати, у всесвітній павутині можна знайти інформацію на будь-яку тему і з великою ймовірністю, вона буде перекладена англійською мовою.

Багато людей у нашій країні вже дуже добре володіють мовою, тому англійська мова дуже популярна: вона вивчається в школах (іноді навіть в дитячих садах), коледжах і університетах.

Знати англійську мову сьогодні є абсолютною необхідністю для кожної освіченої людини та гарного фахівця. Вивчення мови є не легкою справою. Це довгий і повільний процес, який забирає багато часу і терпіння. Але це потрібно.

Проте як сучасній людині вивчити англійську та отримати можливість підкорити цей світ?

Завдяки розвитку технологій кожен охочий може вивчити англійську просто вдома. Для цього існують різноманітні способи, такі як, спеціальні сайти, додатки на

мобільний телефон та інші. Та кожен з них має велику кількість недоліків, що роблять процес надто повільним, нудним та не продуктивним.

Для тих хто хоче особистого спілкування з самим викладачем та групою студентів існує не мало курсів, тренінгів та вебінарів. Але на жаль більшість з них є недешевими, що лякає більшість учнів.

На щастя в нашій країні стрімко зростає потреба у вивченні англійської мови та через це навчання другої іноземної мови вже є обов'язковим в університетах та школах. Проте більшість викладачів погодиться з тим, як складно викладати англійську мову в групі з 20 та більше студентів.

Знання англійської мови, безумовно, важливе в сучасному суспільстві, але скоріш за все ці знання студент отримає в якійсь спеціалізованій школі вивчення зарубіжних мов, тому що викладач не може дати такий урок, щоб кожен з 20-30 студентів зрозумів абсолютно весь матеріал. Але кожен студент хоче прийти на заняття і показати свої успіхи у вивченні якоїсь теми, відповідати під час заняття та вдосконалювати свої навички спілкування англійською мовою та подолати «мовний бар'єр».

В такому випадку у студента може зникнути бажання вивчати англійську мову, оскільки у викладача просто немає можливості виділити усім час та попрацювати індивідуально з кожним учнем, що є важливим.

Саме через ці проблеми та таку необхідність вивчення іноземної мови мною було проведене дослідження, що сприяло розробці веб-додатку, тобто сайту, який допоможе розв'язувати питання нудного, складного та не індивідуального підходу у навчанні в університетах та інших навчальних закладах. Також, веб-додаток, допоможе у навчанні за умов, коли учням та вчителям потрібно знаходитись вдома, що як ніколи актуально в наш час.

1. ПОСТАНОВКА ЗАДАЧІ

Задача диплому — реалізувати веб-додаток для ефективного вивчення англійської мови, за допомогою якого можна персоналізувати розвиток вивчення англійської мови, серед студентів вищих навчальних закладів. Dodatok повинен запропонувати оптимальну стратегію боротьби зі складним та не індивідуальним підходом вивчення англійської мови.

В результаті розробки програмного продукту, викладачу з англійської мови буде надана можливість створювати завдання у вигляді тестів з граматики, гри для збільшення словникового запасу та практик розмовного англійського. Далі викладач має можливість опрацьовувати дані студентів, тобто перегляд їхньої успішності, кількість набраних балів за тести та виставлення оцінок відносно цих показників. Студенту, у свою чергу, надається доступ до проходження створених викладачем різноманітних тестів, перегляд особистого словника та запис нових слів в нього, а також надається можливість переписуватись зі студентами своєї групи в розділі чату.

Розроблена система повинна забезпечувати наступні можливості:

- Можливість реєстрації для будь-якого користувача
- Для раніше зареєстрованих користувачів в системі передбачена форма для авторизації.
- Внутрішній зміст домашньої сторінки має містити блоки: Профіль, Навігаційне меню, Тести, Словник, Тренування, Чат.
- Пункт Тест повинно відкриватися вікно тестів з різними можливостями в залежності від ролі користувача.
- Студенту повинна бути доступна функція проходження тесту та отримання статистики щодо успішності.
- Викладачу повинна бути доступна функція створення\редагування \видалення тестів.

- При виборі пункту Словник (доступне лише студенту) повинно відкриватися вікно словника, яке містить такі блоки як самі слова, функція додавання слова та набори слів.
- Пункт Матеріали повинен містити вікно з доступними матеріалами для вивчення.
- Пункт Тренування повинен містити вікно з наявними вправами для проходження.
- Пункт Чат повинен містити область листування та список вже розпочатих чатів.

Мета цього проекту полягає в тому, щоб дати змогу студентам вищих навчальних закладів, з будь-яким досвідом користування комп'ютером, можливість вивчати англійську мову легко та цікаво. Викладачам, у свою чергу, дає можливість надавати матеріал для вивчення та слідкувати за успішністю своїх студентів навіть за умов самоізоляції.

Для більш детальної постановки задачі мені потрібно проаналізувати потрібність англійської мови у сучасному світі, проаналізувати основні проблеми у сучасному навчанні та розглянути вже існуючі способи вивчення англійської мови, що я зробив у розділах 2.1 - 2.6. У розділі 2.7 - 2.8 я детально описую основні вимоги до програмного продукту та окремо описую функціональні вимоги до кожного з розділів мого проекту.

2. АНАЛІЗ ПРОБЛЕМИ ВИВЧЕННЯ АНГЛІЙСЬКОЇ МОВИ У СУЧАСНОМУ СВІТІ

2.1 Загальні поняття та скорочення

База даних — це організаційна структура, призначена для зміни, зберігання та обробки взаємозалежної інформації, зазвичай великих обсягів. Бази даних часто використовуються для динамічних сайтів зі значними обсягами даних - часто це інтернет-магазини, портали, корпоративні сайти.

Skype — це програмне забезпечення для зв'язку зі світом. За допомогою Skype мільйони людей і компаній можуть здійснювати індивідуальні та групові голосові й безкоштовні відеодзвінки, а також відправляти миттєві повідомлення і файли іншим користувачам Skype. Ви можете користуватися Skype на будь-якому зручному для вас пристрої: на мобільному телефоні, комп'ютері або планшеті.

Інформаційні технології — це галузь, яка відповідає за збір, зберігання і передачу інформації за допомогою технічних пристроїв і, в цілому, за спілкування людей на відстані. Справжній бум інформаційних технологій відбувся в 1990-і в зв'язку з поширенням персональних комп'ютерів, мережі інтернет і персональних мобільних телефонів. Сьогодні в Росії приблизно 89% молодих людей постійно користуються інтернетом і мобільними телефонами. Інформаційні технології (або IT, Information Technology) - самий бурхливо розвивається сектор світової економіки.

Програмний продукт — це комплекс взаємопов'язаних програм для розв'язання певної проблеми (завдання) масового попиту, підготовлений до реалізації як будь-який вид промислової продукції.

Програмні продукти можуть створюватися як індивідуальна розробка під замовлення, так і як розробка для масового поширення серед користувачів.

Веб-додаток являє собою веб-сайт, на якому розміщені сторінки з частково або повністю несформованим вмістом. Остаточний вміст формується тільки після того, як відвідувач сайту запросить сторінку з веб-сервера. У зв'язку з тим що остаточний

вміст сторінки залежить від запиту, створеного на основі дій користувача, така сторінка називається динамічною.

Спектр використання веб-додатків дуже широкий. В даному розділі розглядаються загальні питання використання веб-додатків, а також наводиться приклад простого веб-додатку.

Чекбокс (від англ. Check box), галочка — елемент графічного інтерфейсу, що дозволяє користувачеві управляти параметром з двома станами - ☒ включено і ☐ відключено. У включеному стані всередині чекбокса відображається позначка (галочка (✓), або рідше хрестик (×)). За традицією прапорець має квадратну форму. Поруч з прапорцем відображається його позначення, зазвичай - підпис, рідше - значок. Для збільшення площі активного елемента зазвичай однаково реагує на натискання мишею як на сам прапорець, так і на підпис до нього.[7]

Акаунт (англ. Account) означає «обліковий запис» і являє собою набір даних про користувача, які той вводить і зберігає на будь-якому сайті або інтернет-сервісі.

Іншими словами, акаунт - це інтернет-паспорт, який користувач заповнює для реєстрації на необхідному сайті.

2.2 Роль англійської мови у сучасному світі

На нашій землі існує близько 200 країн, а мов якими розмовляють їх жителі, більше ніж кілька тисяч. Звідси виникає питання, як населенню Землі зрозуміти один одного? Саме через це питання, виникли міжнародні мови, які й дозволяють всім жителям землі спілкуватись один з одним, незалежно від національності та місця локації. Однією з таких міжнародних мов і є англійська. Більш того ця мова вже стала міжнаціональною мовою спілкування для всієї земної кулі.

Важко переоцінити значення англійської мови в сучасному світі. Тому що, це вибір більше ніж 1 мільярда людей, які ним користуються. З них понад 450 мільйонів людей вважають її рідною. Ще 600-650 мільйонів громадян використовують англійську в якості додаткової мови для спілкування. Звичайно, діапазон популярності англійської мови в сучасному світі настільки великий, що через це, англійська перестала бути

ідентичною у різних областях. Навіть якщо, не брати до уваги різноманітні її варіанти та наявність особливостей для кожної національності, англійська мова залишається найпопулярнішою на нашій земній кулі. Але звідки ж з'явилася англійська мова?

З'ясувати точно, як утворилася мова, майже неможливо. Відомо, що походження англійської мови невідривно пов'язане з історією Англії, яка бере свій початок в п'ятому столітті нашої ери. Саме в той час, в Британію вторглися три німецьких племені. Населення країни в цей період становили кельти та римляни, проте німецький вплив було настільки сильним, що незабаром від кельтського і латинської мов на території Англії нічого не залишилося. Ще через деякий час колонію захопили французи, нескладно уявити які видозміни зазнала в цей період англійська мова.

Всі ці перипетії, безсумнівно, внесли зміни в походження англійської мови, в її розвиток і становлення. За кілька років її словник збільшився майже вдвічі, увібравши в себе чимало слів французького та англійського походження, які й до цього дня використовуються в лексиконі справжніх англійських громадян.

Яку роль грає англійська мова в житті людини зараз?

В цілому, економічне, політичне, спортивне та наукове життя нашої планети відбувається англійською мовою. Англійська уже давно визнана офіційною та робочою мовою ООН. Підписання законів і указів, різноманітні зустрічі та саміти глав держав, дебати і переговори - все це проходить англійською мовою. Робота міжнародної банківської системи, торгівля, діяльність транспортної системи в повітрі, на суші та на морі можлива завдяки англійській мові. Саме ця мова є живим інструментом спілкування вчених усього світу, навіть олімпійські ігри і змагання між країнами вибрали офіційною мовою саме англійську.

Знання англійської мови в сучасному світі вже не є чимось особливим, тепер це необхідність. Її вивчають по всій землі: в школах, іноді навіть в дитячих садах, коледжах, університетах та курсах, а в наш час цифрових технологій, будь-хто бажаючий може вивчати англійську навіть по Skype, не виходячи з дому.

Вивчення англійської мови, як потреба, очевидна студентам, які бажають отримати хорошу освіту. Її вивчення дозволяє навчатись в будь-якій сфері, адже саме англійською мовою написанні більшість довідників, книг та документаций по різних темах. Завдячуючи знанню міжнародної мови можна поїхати на навчання в найкращі університети у різних країнах. Саме для подорожей або навіть переїзду до інших країн, більшість людей і вивчають англійську мову.

Вивчення англійською мови грає важливу роль у багатьох сферах. Більшість роботодавців готові більше платити робітникам, які володіють міжнародною мовою. В деяких компаніях на роботу беруть тільки тих співробітників, які володіють іноземною мовою.

Вивчення англійської мови завжди є основною потребою, у великих компаніях, при прийомі на ту чи іншу посаду. Деякі фірми готові своїм коштом навчати перспективних співробітників. Знання іноземної мови також потрібно для менеджерів, оскільки у великих компаніях вони контактують з іноземними партнерами. Не обійтися без іноземної й працівникам туристичної сфери. Проте одною з основних професій де потрібна англійська мова, являється програміст.

2.3 Роль англійської мови у житті програміста

Наскільки ж програмісту важливо знати англійську мову? Чи можна обійтися без цього? Можна, але зарплата в таких випадках менша (більш, ніж вдвічі). Якщо цього аргументу буде мало, ось ще кілька аргументів:

- Всі базові (і не тільки) мови програмування ґрунтуються на англійській мові. На практиці це означає, що освоїти нову мову програмування і навчитися більш віртуозно поводитися з уже знайомими буде значно простіше, якщо Ви володієте англійською.
- Вся фундаментальна спеціалізована література написана, в першу чергу, англійською мовою. І далеко не все було перекладено на українську чи російську мову. Ще менше матеріалу переведено коректно.

- В роботі нерідко виникають труднощі. Іноді доводиться шукати розв'язок проблеми у всесвітній павутині, і часто буває так, що потрібної інформації, знайомою мовою, просто не публікувалося. А ось в англomовному середовищі можна знайти практично будь-яку необхідну інформацію, особливо по темі IT.
- Всі найбільші IT компанії світу базуються в США (Google, Microsoft, ORACLE, Apple etc.) і навіть в їх російських підрозділах переважає англійська мова. Так що, якщо ви не володієте мовою, то побудувати кар'єру в одній з найбільших світових IT-корпорацій у вас не вийде.
- Програмістам нерідко доводиться працювати з закордонними замовниками. Найчастіше вся комунікація відбувається виключно англійською - і первинні переговори, і складання технічного завдання, і його узгодження, і взаємодія під час реалізації проекту та інші.

2.4 Способи вивчення англійської мови

У сучасної людини, яка прагне розвиватись, все більше і більше виникає потреба вивчити англійську мову, і як ми знаємо, чим більший попит на вивчення, тим більше з'являється можливостей. На даний момент є безліч способів вивчення іноземних мов. З розвитком технологій та інтернету з'явилась можливість вивчати англійську не виходячи з дому. Та кожен зі способів базується на певному методі вивчення, що розвиває певні напрями, такі як розмовний англійський, граматики чи аудіювання. В цьому розділі я спробую розібрати найпопулярніші з них.

○ Вивчення у навчальних закладах

Слід почати з того, що сучасна освітня програма охоплює обов'язкове вивчення англійської мови, іноді навіть з дитячого садка. Шкільні заняття складаються з певної програми за якою і слідує викладачі, та на кожен нову тему відводиться визначена кількість годин, але при цьому не завжди всі учні встигають засвоїти матеріал. Кожен викладач знає, як важко викладати для групи яка складається більше ніж з 20 людей, кожному студенту потрібен індивідуальний підхід, та часу надто мало.

В такому випадку у студента може зникнути бажання вивчати англійську мову, оскільки у викладача просто немає можливості виділити усім час та попрацювати індивідуально з кожним учнем, що є важливим. З цього можна зробити висновок, що не кожен учень знає англійську після проходження шкільної програми. З цими знання учень йде до університету, де повторюється та ж сама ситуація.

○ **Курси та тренінги**

Проте більшість надають перевагу заняттям в традиційній обстановці, де є викладач та студенти, що дають можливість активно взаємодіяти. Тому якщо студент не має можливості зробити це в школі чи університеті, то він звертається на курси або тренінги.

На даний момент курси це один з найпопулярніших видів вивчення англійської мови. З'явилися навіть різні види курсів, які роблять більш ухил на ту чи іншу область вивчення. Так, наприклад, з'явилися Speaking club (Розмовні клуби), де студентам дають можливість поговорити на ту чи іншу тему англійською мовою.

Крім того, є велика кількість тренінгів, які вже давно перейшли в формат онлайн, де студенту надається вся потрібна інформація в ході його вивчення. Мінусом є те, що немає чіткого контролю з боку більш досвідчених викладачів, також більшість тренінгів чи курсів являються платними.

○ **Репетитор**

Проте більшість курсів чи тренінгів являються платними, і навіть якщо не враховувати це, найголовнішим недоліком є те, що викладачі на курсах все ще не мають можливості індивідуально приділити увагу кожному студенту. Та для цього існують репетитори.

Напевно найефективніший спосіб вивчення англійської, це репетитор, так як він приділяє вам індивідуальний підхід і може допомогти вам підтягнути саме ту тему в якій ви менше розбираєтесь. Через те, що не всі репетитори мають ідеальний розмовний англійський і допомагають вони в основному для вивчення граматики та підготовки до різних екзаменів таких як ЗНО, ДПА та інші

Проте на даний момент є можливість не тільки особистого спілкування офлайн, а також є багато сервісів, де можна підібрати собі викладача і спілкуватись з ним онлайн. Також на таких сайтах можна знайти навіть носія мови.

○ **Живе спілкування з носіями мови**

Як правило спілкування з носієм мови є найефективнішим способом для розвитку розмовного англійського, що може допомогти в подорожах чи наприклад здивувати колег на роботі.

Та як можна поспілкуватись з носієм? Є різні способи, один з них це подорожі, де можна познайомитись з людьми в публічних місцях, як правило люди там дуже відкриті до діалогу, завдяки цьому можна не тільки прокачати свій англійський, а і знайти нові знайомства.

Також існують різноманітні сайти, де можна знайти носіїв тої чи іншої мови та поспілкуватись з ними в режимі онлайн.

○ **Книжки**

Часто книжки використовуються як додатковий матеріал для більш тісного знайомства з мовою. Для тих, хто вивчає мову самостійно, книги та самовчителі можуть стати чудовим засобом освоєння англійської мови в прийнятному темпі, без обмежень і зобов'язань, які присутні в інших формах навчання. Проте не всім підійде подібний формат.

○ **Фільми, серіали та відео з субтитрами**

Перегляд улюблених серіалів чи фільмів може допомогти у розширенні словникового запасу та вивченні сталих фраз якими користуються носії мов у реальному житті. Та під час перегляду можна ненароком захопитись своїм улюбленим серіалом та забути, що вивчаєш мову, і пропустити багато важливих нових слів.

Також можна переглядати відео з цікавими для вас темами англійською мовою, що допоможе не тільки дізнатись більше у вибраній темі, а збільшити словниковий запас, саме тими словами які вам будуть потрібні для підтримання бесіди у цій темі

Однак, відсутність спілкування з викладачем або іншими студентами може створити проблеми. У разі, якщо навчання дається важко і складно просунутися далі. Відеокурси можуть відмінно підійти тим, хто вже володіє базовим матеріалом, але хоче оновити знання.

○ Аудіоуроки

На даний момент цей метод поступово застаріває, тому що, багато хто приходить до висновку, що використовуючи один лише цей метод, мову освоїти складно. Однак, хороші онлайн-курси англійської мови нерідко пропонують аудіо матеріали, щоб дати більш повне уявлення про вимову ти діалекти.

Аудіозаписи часто використовуються як доповнення до навчальної програми, однак, існують і повні аудіоуроки для тих, кому легше вивчати мову на слух.

○ Онлайн-навчання

Метод онлайн-навчання добре зарекомендував себе як серед викладачів, так і серед тих, хто вивчає мову. Володіючи гнучкістю і зручністю, він добре підійде тим, хто не може підтримувати графік занять в класі, але вміє працювати в групах та індивідуально. Маючи уявлення про різні методи вивчення англійської, ви зможете вибрати найбільш сприятливий для вас.

Через розвиток технологій, така річ як мобільний телефон перетворилась з простого способу зв'язку з людьми, у міні-комп'ютер який несе в собі найрізноманітніші функції. Все більше можливостей розробники вкладають у звичайний мобільний телефон, не винятком стало і вивчення англійської мови. На даний момент є безліч додатків, що замінюють більшість способів вивчення. Про них поговоримо в наступному розділі.

2.5 Моделі додатків для вивчення англійської мови

У попередньому розділі ми поверхнево пройшли по основним способам вивчення англійської мови, у цьому ж розділі я розібрав найпопулярніші та найефективніші

моделі онлайн вивчення, що вже існують, та виділив основні переваги та недоліки кожної з них.

Почнемо з найбільш розповсюдженого способу дистанційного вивчення англійської мови – веб-додатки, які надають можливість навчання в інтерактивній формі. Як правило, вони платні, проте користувач може використовувати базові функції сервісу без оплати, хоча для доступу до найцікавішого матеріалу необхідно вносити плату. Але в додаткових платних можливостях вже є добірка різноманітних матеріалів відповідно до рівня знань користувача, можливість розширення словникового запасу, можливість тренування читання. Деякі сервіси пропонують і вивчення граматики, як частину доповненого (платного) функціоналу. Найбільш відомими представниками таких сервісів є всім відомі LinguaLeo та Puzzle English.

Головною особливістю LinguaLeo можна виділити стилістику сайту - на ресурсі «живе» герой лев Лео, якого необхідно кожен день годувати фрикадельками (Своєрідний показник прогресу у навчанні; видаються за певні успіхи у навчанні). Ресурс також надає можливість читати різноманітні тексти, прослуховувати аудіо. Цей сервіс підходить для користувачів з початковим рівнем знань, і має обмежену структурованість навчання, бо користувачу пропонуються випадкові завдання, котрі ніяк не зв'язані між собою.

Інший схожий сервіс Puzzle English, що являє собою поєднанням різних способів вивчення, таких як, завдання для сприйняття на слух, перегляд серіалів із субтитрами тощо. Але цей ресурс також орієнтований на користувача з початковим рівнем знань і також ніяк не структурований.

Також слід згадати про безкоштовний додаток для мобільного телефону Dualingo. Додаток дуже простий в користуванні. В ньому реалізовані методи вивчення граматики шляхом тестів, які передбачають завдання на прослуховування речень на англійській та навіть говоріння. З плюсів хочеться зазначити рейтинг між користувачами, що дозволяє переходити в вищі ліги, шляхом набирання балів за

тести. З мінусів, всі тести досить поверхневі та дуже повільно збільшують словниковий запас.

Друга модель вивчення мови полягає у використанні персональних уроків з викладачем по скайпу, розглянемо на прикладі ресурсу SkyEng. Такий метод має безліч плюсів – це зручно, викладач мотивує учня пропонуючи цікаві завдання та вправи, сучасні теми, а не застарілі підручники. Незаперечним мінусом такого типу навчання є дуже висока ціна – не кожен може виділити досить велику суму необхідну на оплату повного періоду навчання, також тяжко підібрати зручний час для заняття з незнайомою для вас людиною.

Третій метод націлений на запам'ятовування слів шляхом інтервальних повторювань. Найкращим додатком такого типу є Memrise. Показуючи кожне слово, програма пропонує користувачу поставити відмітку: легко, складно, вивчено. І вже в наступні рази додаток пропонує завдання пов'язані з конкретним словом в певні проміжки часу, в наслідок чого більш складні слова користувач повторює частіше.

Вище було розглянуто декілька основних моделей вивчення англійської мови з використанням ресурсів Інтернету. Кожен з них має ряд переваг та недоліків. Підсумуємо цю інформацію у вигляді таблиці.

Таблиця 2.1 – Огляд ресурсів для вивчення англійської мови

| Веб-додаток | LinguaLeo | SkyEng | Memrise | Puzzle-English | Dualingo |
|---------------------------------|------------|------------|------------|----------------|------------|
| Платні функції | + | + | + | + | - |
| Рівень знань користувача | початковий | початковий | початковий | початковий | початковий |
| Структурована методика навчання | - | - | + | - | + |
| Відеоматеріали | частково | частково | - | частково | - |
| Занурення в англomовну середу | - | + | - | частково | - |
| Тренування написання | + | - | частково | - | частково |
| Мотивація користувача | герой Лео | - | рейтинг | - | рейтинг |

2.6 Виділення основних проблем

У попередніх розділах я спробував проаналізувати основні види вивчення англійської мови, після чого вибрав найефективніші з них та знайшов вже існуючі реалізації цих методів. Дослідивши кожен з них можна помітити основний мінус, більшість розширених функцій для вивчення являються платними. Та навіть не враховуючи це в кожного способу вивчення є ще кілька недоліків.

Визначимо дерево проблем:



Рисунок 2.1 – Дерево проблем

Завдяки визначенню основних проблем найпопулярніших сервісів для вивчення англійської мови, з'явилась можливість сформулювати дерево цілей(рисунки 2.2), що допоможе краще орієнтуватись під час створення нового веб-додатку.

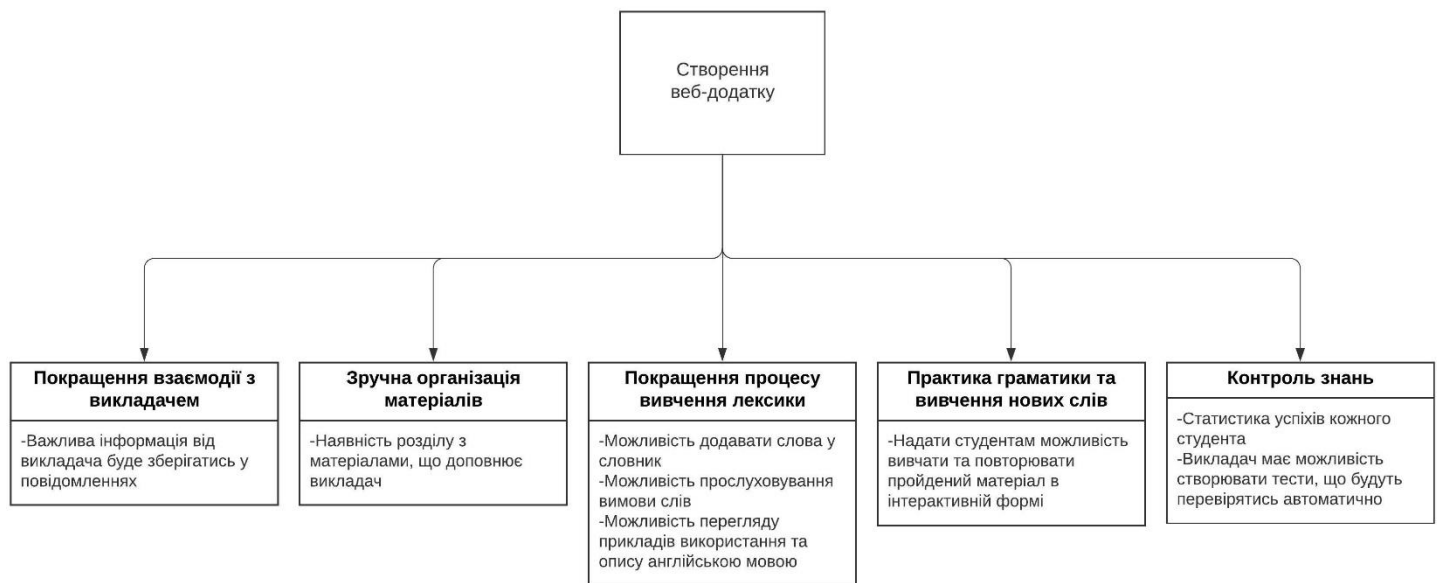


Рисунок 2.2 – Дерево цілей

2.7 Визначення основних вимог до ПП

Завдяки дереву цілей, що ми визначили в попередньому розділі, ми можемо сформулювати те, як буде виглядати наш веб-додаток.

Наш ПП буде містити:

- Тести
- Словник, який містить набори слів
- Тренування слів зі словника
- Чат для спілкування студентів та викладача

Щоб краще сформулювати вимоги до кожного з розділів, намалюємо діаграму використання ПП.

Користувача буде зустрічати початкова сторінка, де можна авторизуватись в системі, або зареєструватись після чого увійти в систему. Після чого система може обіграти два сценарії, де користувач являється викладачем або користувач являється студентом.



Рисунок 2.3 – Діаграма початкової сторінки

В наступній діаграмі (рисунок 2.4) представлений сценарій, де користувачем являється викладачем та зображенні основні можливості, які є тільки у викладача.

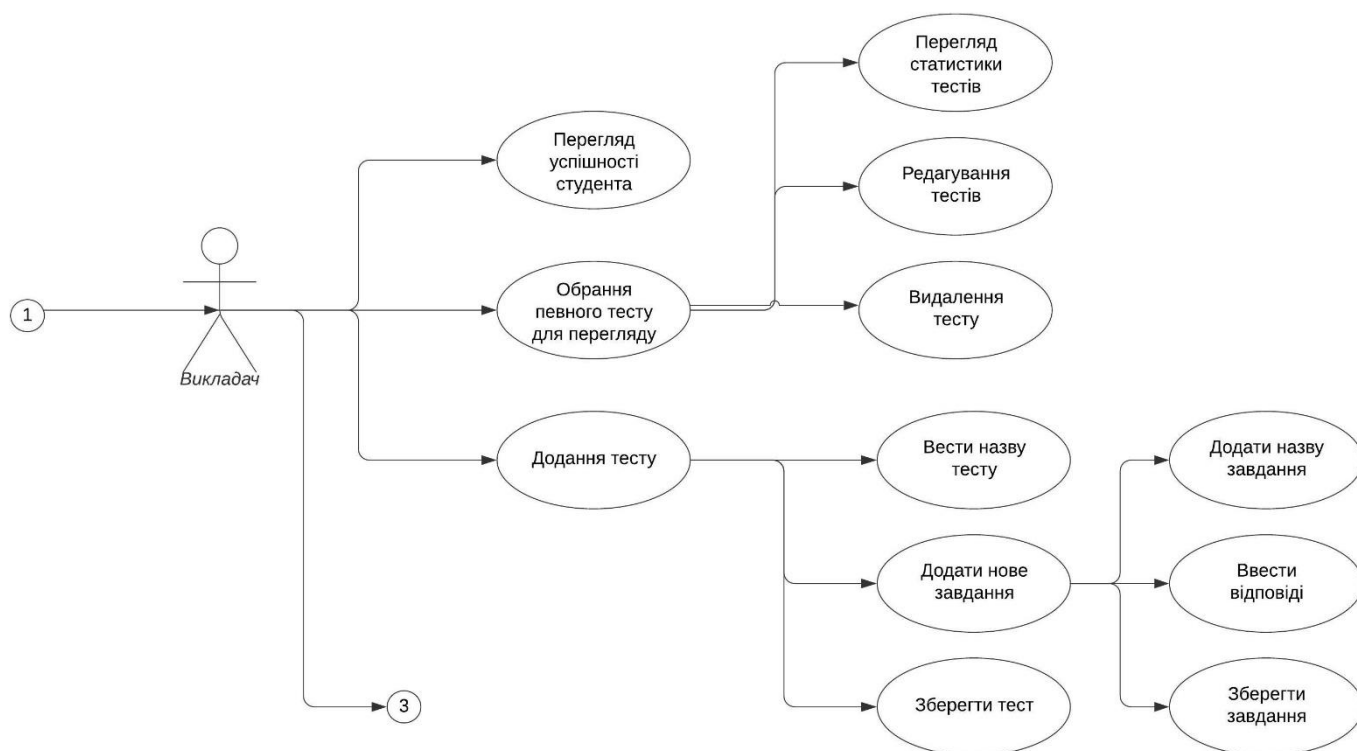


Рисунок 2.4 – Діаграма викладача.

Також є деякі можливості, які доступні як для викладача, так і для студента. Наступна діаграма (рисунок 2.5) зображає ці можливості.

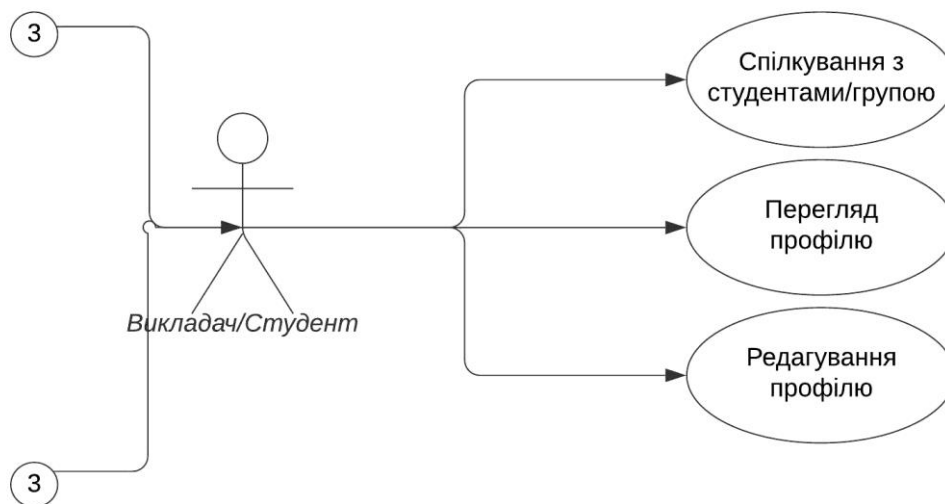


Рисунок 2.5 – Діаграма спільного доступу.

На наступній діаграмі (рисунок 2.6) представлений сценарій, де користувачем являється студент та зображенні основні можливості, які є тільки у студента.

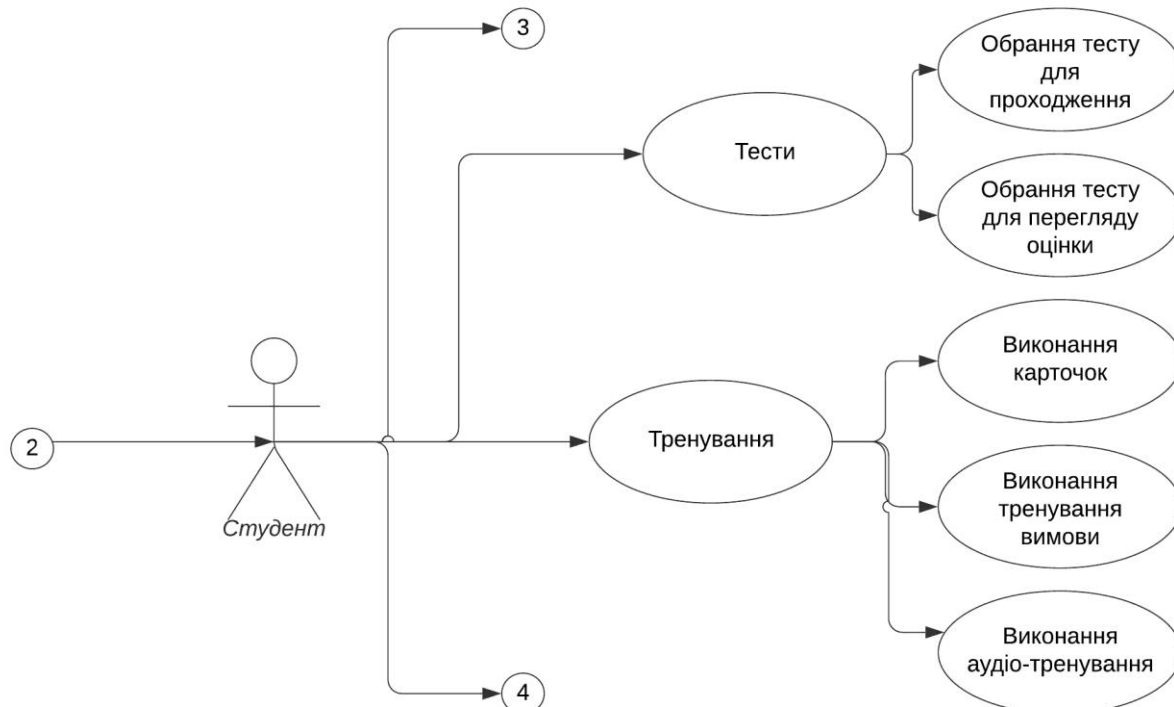


Рисунок 2.6 – Діаграма студента.

Додаткові можливості, словник та матеріали зображені на діаграмах (рисунок 2.7-8).



Рисунок 2.7 – Діаграма студента(словник)

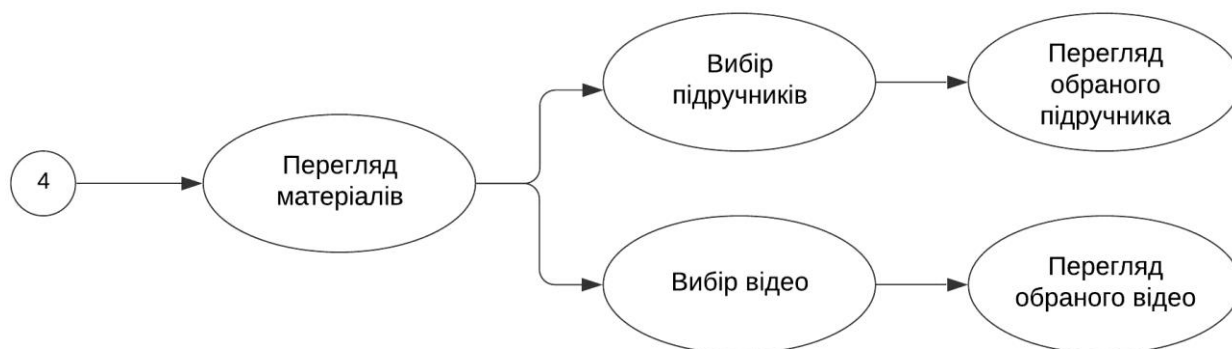


Рисунок 2.8 – Діаграма студента(матеріали)

З допомогою діаграм задаємо основні вимоги до інтерфейсу:

- Веб-додаток повинен надавати можливість реєстрації для будь-якого користувача.
- Реєстрація повинна бути реалізована шляхом форми, яка містить введення електронної пошти, паролю тощо.
- Для раніше зареєстрованих користувачів в системі передбачена форма для авторизації.
- Авторизація повинна бути реалізована шляхом форми, яка містить введення електронної пошти та паролю.

- Кожен авторизований користувач повинен мати можливість вийти з системи для виключення непередбачуваних дій, які можуть бути здійснені іншими користувачами комп'ютера.
- Кожен авторизований користувач повинен мати налаштування для зміни особистих даних.
- При вході на сайт користувачу повинна відкриватися домашня сторінка з наявним функціоналом.
- Внутрішній зміст домашньої сторінки повинен містити такі блоки як профіль, навігаційне меню з такими пунктами, як Тести, Словник, Тренування, Чат.
- При виборі пункту Тести повинно відкриватися вікно тестів з різними можливостями в залежності від ролі користувача.
- Студенту повинна бути доступна функція проходження тесту та отримання статистики щодо успішності.
- Викладачу повинна бути доступна функція створення\редагування\видалення тестів..
- При виборі пункту Словник (доступне лише студенту) повинно відкриватися вікно словника, яке містить такі блоки як самі слова, функція додавання слова та набори слів.
- При виборі пункту Матеріали повинно відкриватися вікно з доступними матеріалами.
- При виборі пункту Тренування повинно відкриватися вікно з наявними вправами.
- При виборі пункту Чат повинна відкриватися область листування та список вже розпочатих чатів.

Складемо ТЗ - функціональні вимоги до кожного з розділів:

Розділ Реєстрації\Авторизації\Комунікації (SignUp\Register\Chat)

- Поля форми Register та SignUp, а саме Email, Password, Name and Second name (до 20 символів в кожному), Phone (13 символів), Group - повинні являти собою поля для вводу тексту, а поле вибору Role має являти собою список, що розкривається, та містить у собі два значення student/teacher.
- Пароль шифрується за алгоритмом md5 при передачі у базу даних.
- Поле вводу повідомлення являє собою поле для вводу тексту.
- При натиску на кнопку «Send» або Enter, повідомлення відправляється в чат, де стає доступним для інших користувачів.
- При натиску на користувача з чату відкривається його профіль, де можна почати з ним особистий чат шляхом натиску на кнопку «Send a message».

Розділ **ТЕСТИ (Tests)** містить такі підрозділи, як

- Перегляд всіх тестів після вибору передбачає відображення усіх можливих тестів, які користувач може виконати, сортування тестів відбувається за датою додавання тесту
- «Create» (для викладача) після вибору передбачає відкриття форми з таким полем, як «Set name of Test», кнопками «Add Task» та «Save».
 - Поле «Set name of Test» має обмеження на кількість символів (до 50 символів) та написання тільки латинських літер та цифр, без спеціальних знаків.
 - Кнопка «Add Task» після натискання відкриває міні-форму для додавання нового завдання з таким полем, як «Set Task», чекбокс з варіантами відповідей на запитання,
 - Поле «Set Task» містить обмеження на кількість символів (до 200) та написання тільки латинськими літерами та цифрами з спеціальними символами.
 - Кнопка «Add Answer» відкриває поле «Answer» для введення відповіді на запитання. Поле має обмеження в 200 символів.
 - В чекбоксі при створенні тесту вказується правильна відповідь на запитання.
 - Кнопка «Save» передбачає збереження тесту у відповідній таблиці бази даних.

При виборі певного тесту для виконання\перегляду, в залежності від типу акаунту (студент\викладач) доступні такі функції:

для студента:

- тест має список питань з вибором відповідей та кнопку для завершення тесту
- при завершенні виконання тесту, якщо якесь питання залишилося без відповіді, буде отримано відповідне попередження, та тест буде неможливо закінчити, поки не буде дано відповіді на кожне питання
- при успішному завершенні тесту система проаналізує кількість правильних відповідей та видає повідомлення про правильність виконання тесту у відсотках
- якщо тест вже був виконаний, в списку всіх тестів, він буде позначений як «Complete», а при виборі пройденого тесту користувач отримує інформаційне повідомлення про успішність виконання тесту (попередній пункт).

для викладача:

- тест має список питань з вже обраними відповідями та кнопки такого призначення, як «Edit», «Delete», «Statistic».

- «Edit» передбачає відкриття вікна створення тесту перед тим, як він був збережений.
- «Delete» передбачає повне видалення тесту з бази даних.
- «Statistic» показує список всіх учнів, які виконали даний тест з відсотком правильних відповідей.

Розділ **СЛОВНИК(Vocabulary)** (доступний тільки для студентів) містить такі підрозділи, як:

- Перегляд всіх слів у словнику
- Пошук слова у словнику
- Перегляд всіх створених наборів слів
- Додавання нового набору шляхом натискання кнопки «Add Set».
- Додавання нового слова у словник з такою формою:
 - Поле «Ukrainian» містить обмеження на кількість символів (до 20 символів)
 - Поле «English» містить обмеження на кількість символів (до 20 символів)
 - Поле «Definition»
 - При натисканні на кнопку «Add word» введені дані додаються до відповідної таблиці бази даних.

Розділ: **Тренування (Train)**

Даний розділ доступний тільки для студентів. Містить тренування в інтерактивній формі слів\граматики\перевірка мови які охоплюють такі вправи, як робота зі словесними картками, вибір правильних відповідей, аудіювання. В тренуванні використовується особистий словник студента.

Завдання:

«Flash Cards» : перед користувачем набір карточок, необхідно знайти відповідності. При натиску на карточку вона стає активною, очікується вибір наступної карточки користувача. При виборі наступної карточки, виконується перевірка на відповідність. Якщо умова виконана – вони зникають з екрана. Якщо ні – картки стають неактивними. Тренування продовжується доки не будуть обрані всі відповідні картки. Тренування складається з декількох «рівнів» (від 2-х до 4-5) для, по можливості повного, закріплення інформації. (наприклад, 1 - перші 5 слів, 2 - другі 5 слів, 3 - все слова разом.)

Критерії відповідності:

1. Слово та переклад;
2. Слово та його опис;

3. Слово та його вимова;
4. Слово та візуальна асоціація;

«Тренування на слух» : користувач прослуховує аудіо довжиною в 1-2 хвилини або пісню, також його доступний повний текст запису (тільки під час прослуховування), далі він виконує 1-2 з наступного списку: вставити пропуски в тексті, виправити помилки написання, розставити слова в правильному порядку, перевести деякі слова з тексту, розставити частини тексту в правильному порядку. Користувач може прослуховувати декілька разів аудіо, ставити на паузу, промотувати її.

Система визначає наявність помилок і дає можливість їх виправити.

«Перевірка вимови» : користувач має можливість записувати власну вимову та після чого може прослуховувати та перевіряти, чи правильно він сказав те чи інше. Користувачу може надаватись готовий текст для читання та вимови, та записаний його викладачем правильний приклад вимови цього тексту.

2.8 Висновки до розділу 2

У другому розділі було розглянуто аналіз проблеми вивчення англійської мови у сучасному світі та необхідність вивчення англійської мови окремо для програміста. Після впевненості у необхідності вивчення, було розглянуто основні види вивчення англійської мови.

У розділі описано аналіз проблеми найпопулярніших та вже існуючих модулів для навчання та сформовано дерево проблем та дерево цілей. Для зображення можливостей веб-додатку, були створені діаграми, що показуються можливості користування ПП. Завдяки загальному аналізу у розділі, були сформовані основні вимоги до інтерфейсу та функціональні вимоги до кожного з розділів, що допоможе у реалізації веб-додатку.

3. ЗАСОБИ РОЗРОБКИ

3.1 PHP - мова сценаріїв

Програма на PHP відповідає за повернення файлу в чистому вигляді, придатному для зображення в браузері. У найпростішому випадку на виході документа PHP буде тільки код HTML. PHP відноситься до дуже слабо типізованих мов. Це означає, що змінні не вимагають оголошення перед своїм використанням і що PHP завжди перекладе змінні в той тип, який потрібно для їх оточення на момент доступу до них.

За замовчуванням в кінці імен PHP-документів ставиться розширення PHP. Коли веб-сервер зустрічає в файлі це розширення, він автоматично передає файл PHP-процесору. Веб-сервери мають досить широкий діапазон налаштувань, і деякі веб-розробники вибирають такий режим роботи, при якому для PHP-процесору примусово передаються також файли з розширеннями XML або HTML.

Можливості PHP

Почнемо з того, що PHP використовується для створення скриптів, які працюють на стороні сервера. PHP може вирішувати ту ж роботу, що і будь-який інший CGI-скрипт, у тому числі обробляти дані html - форм або динамічно генерувати html сторінки. Всього виділяють три основні області застосування PHP:

- Створення скриптів для виконання на стороні сервера. PHP зазвичай використовується саме таким чином. Для цього будуть потрібні три речі. Інтерпретатор PHP, веб-сервер і браузер. Для того щоб можна було переглядати результати виконання PHP-скриптів в браузері, потрібен робочий веб-сервер і встановлений PHP.
- Створення скриптів для виконання в командному рядку. Є можливість створити PHP-скрипт, здатний запускатися без сервера або браузера. Все, що потрібно - парсер PHP. Ці скрипти можуть бути залучені в завданнях простої обробки текстів.

- Створення віконних додатків, що виконуються на стороні клієнта. При хорошому знанні PHP і бажанні використовувати деякі його можливості у своїх клієнтських додатках, то можна використовувати PHP-GTK для створення таких додатків. Подібним чином можна створювати і крос-платформні додатки. PHP-GTK є розширенням PHP і не поставляється разом з основним дистрибутивом PHP [8].

Динамічне зв'язування в PHP

Оскільки PHP є мовою програмування й отримана, в результаті його роботи, інформація на виході може бути абсолютно різною для різних користувачів. При кожному натисканні користувача на якомусь елементі подробиці можуть відправлятися назад тією ж веб-сторінці, яка буде приймати рішення, що робити далі, відповідно до різних об'єктів cookie і даних сесії, які можуть бути збережені.

Але попри можливість створення у такий спосіб цілого сайту, цього робити не рекомендується, оскільки вихідний код буде весь час розростатися і набуде величезних розмірів в міру того, як йому доведеться брати до уваги різноманітні дії користувача. Буде набагато краще та розсудливіше розділити розробку сайту на кілька різних частин. Наприклад, один автономний процес буде займатися підпискою на сайт, де будуть враховуватись усі наслідки, що впливають, такі як, перевірка допустимості адреси електронної пошти, незалучені імена користувача та інші.

Другий модуль непогано було б створити для реєстрації користувачів, попереднього їх допуску до основної частини сайту. Потім можна створити модуль виведення повідомлень, в якому користувачі могли б залишати свої коментарі, модуль, що містить посилання і корисну інформацію, ще один модуль, дозволяє завантажувати на сайт фотографії, та інші.

Одним з найбільш популярних в даний час додатків, керованих PHP, є платформа для ведення блогів WordPress. При читанні сайту цього можна і не зрозуміти, але для кожної основної секції виділений свій основний PHP-файл, а величезна кількість

спільно використаних функцій вміщено в окремі файли, які включаються основними PHP-сторінками за потреби.

Вся платформа тримається на закулісному відстеженні сесії, і можна не помітити, коли здійснюється перехід від однієї підпорядкованої секції до іншої. Тому, якщо веб-розробник хоче провести тонке налаштування WordPress, йому не важко знайти конкретний файл, який для цього застосовується, і виконати його перевірку і налагодження, не втрачаючи даремно часу на які не пов'язані з ним частини програми.

Об'єкти PHP

При створенні програми, розрахованої на використання об'єктів, потрібно сконструювати певну сукупність даних і коду, так звану класом. Кожен новий об'єкт, заснований на цьому класі, називається екземпляром (або випадком вживання) цього класу. Дані, пов'язані з об'єктом, називаються його властивостями, а використані функції - методами. При визначенні класу задаються імена його властивостям і код для його методів.

При створенні об'єктів переважно користуються інкапсуляцією або створюють клас таким чином, щоб з його властивостями могли працювати тільки його власні методи. Іншими словами, потрібно заборонити зовнішньому коду доступ до даних об'єкта. Надані об'єктом методи відомі як інтерфейс об'єкта. Такий підхід спрощує налагодження: дефектний код доведеться виправляти тільки в межах класу.

Крім того, коли потрібно буде оновити програму, при використанні інкапсуляції і підтримки однакового інтерфейсу можна буде просто розробити нові класи для заміни старих, повністю їх налагодити, а потім замінити ними старі класи. Якщо вони будуть у чомусь непрацездатними, можна буде повернути назад старі класи для негайного усунення проблеми перед подальшим налагодженням нових класів[1].

3.2 JavaScript - об'єктно-орієнтована мова програмування

JavaScript надає сайтам динамічну функціональність. Тобто якщо при проходженні вказівника миші над елементом браузера що-небудь виділяється, з'являється новий текст, змінюється колірне оформлення або зображення, все це робиться за допомогою JavaScript. Ця мова пропонує такі ефекти, яких не можна досягти ніякими іншими засобами, оскільки він запускається всередині браузера і має безпосередньо доступ до всіх елементів веб-документа.

Оскільки як в JavaScript, так і в PHP підтримуються багато елементів синтаксису структурного програмування, що використовуються в мові програмування C, ці дві мови дуже схожі одна на одну. Обидві відносяться до мов високого рівня. Проте у них досить слабка типізація, що дозволяє легко приводити змінну до нового типу даних лише шляхом застосування її в новому контексті.

Коли HTML-елементи веб-сторінки знайшли більш чітке, структуроване визначення в так званій об'єктній моделі документа – DOM, мова JavaScript отримала ще більші можливості. Об'єктна модель документа дозволила відносно просто додавати новий абзац або сфокусуватися на який-небудь частини тексту і внести в неї зміни.

Об'єктна модель документа

Розробники JavaScript були дуже розумними людьми. Замість того щоб просто створити ще одну мову написання сценаріїв, вони проявили далекоглядність і зробили її навколо об'єктної моделі документа, або DOM (Document Object Model).

Ця модель розбиває частини HTML-документа на окремі об'єкти, у кожного з яких є власні властивості та методи кожен з яких може керуватись за допомогою JavaScript.

В JavaScript об'єкти, властивості та методи поділяються за допомогою точок, саме через це, в якості оператора об'єднання рядків в JavaScript користується знак +, а не крапка. Його методами будуть функції, що займаються витяганням, заміною та іншими діями з властивостями.

Усередині JavaScript вибудовується ієрархія з батьківських та дочірніх об'єктів. Ця ієрархія і називається об'єктною моделлю документа - DOM

Функції, об'єкти та масиви

В JavaScript надається практично такий же доступ до функцій і об'єктів, як і в PHP. Мова JavaScript фактично заснований на об'єктах, оскільки, у нього є доступ до DOM - моделі, яка реалізує доступ до кожного елементу HTML-документа, дозволяючи працювати з ним як з об'єктом. спосіб застосування функцій і об'єктів, а також використовуваний для цього синтаксис дуже схожі на все те, що ми бачили в PHP.

○ Функції JavaScript

Найкраще в JavaScript реалізовані функції. У них майже все правильно. Майже, але не все, як і слід було б очікувати від JavaScript. Кожна функція складається з групи інструкцій. Функції – фундаментальний модульний елемент JavaScript. Вони необхідні для багаторазового використання одних і тих же фрагментів коду, приховування інформації та побудови програм. Функції визначають поведінку об'єктів. Основне завдання програмування - це переклад поставлених вимог до функцій і структури даних.

В JavaScript поряд з доступом до десятків вбудованих функцій (або методів), можна створювати й власні функції. Як тільки з'являється який-небудь складний фрагмент коду з перспективами на багаторазове використання, він стає кандидатом на оформлення у вигляді функції.

○ Об'єкти JavaScript

У порівнянні зі змінними, які в кожен конкретний момент можуть утримувати тільки одне значення, об'єкти JavaScript - це значний крок вперед, оскільки в них може міститися кілька значень і навіть функцій. об'єкт групує разом дані та функції для роботи з ними.

Числа, рядки, логічні змінні (true і false), а також значення null і undefined в JavaScript відносяться до простих типів даних. Все інше є об'єктами. Числа, рядки та логічні змінні схожі на об'єкти тим, що мають методи, але на відміну від об'єктів вони

незмінні. Об'єкти в JavaScript мають змінювані ключові колекції. В JavaScript об'єктами є масиви, функції, регулярні вирази, і, звичайно, об'єкти також є об'єктами.

Об'єкт являє собою контейнер властивостей, де кожне властивість має ім'я і значення. Іменем властивості може бути будь-який рядок, в тому числі порожня. Значенням властивості може бути будь-яке значення, виключаючи `undefined`.

Об'єкти в JavaScript не прив'язані до класів, тому не існує обмежень на іменування нових властивостей або на присвоювання їм значень. об'єкти досить зручно використовувати для збору та організації даних. Об'єкти можуть містити інші об'єкти, тому з їх допомогою можна легко уявити структуру дерева або графа.

В JavaScript передбачена можливість типізації збірки, що дозволяє одному об'єкту успадковувати властивості іншого. При правильному використанні це може скоротити час ініціалізації об'єкта і споживання пам'яті.

- Масиви в JavaScript

Масив являє собою безперервний фрагмент пам'яті, доступ до елементів якого здійснюється за допомогою цілих чисел (індексів), що використовуються для обчислення зсуву. Масиви можуть забезпечувати швидкий доступ до даних.

На жаль, в JavaScript немає нічого схожого на таке визначення масиву. Однак в JavaScript є об'єкти, які схожі на масиви. Індеси масиву перетворюються в рядки, які використовуються для створення властивостей об'єкта. Хоча такі структури значно повільніше реальних масивів, вони можуть виявитися зручніше у використанні. Масиви мають власний формат літералу. Крім того, існує досить великий набір корисних вбудованих методів масивів.

За допомогою літералів масивів досить зручно задавати нові значення масиву. Літерал масиву являє собою пару квадратних дужок, яка містить кілька значень, розділених комами, або не містить взагалі ні одного значення. Літерал масиву може вказуватися в будь-якому вираженні. Перше значення всередині дужок отримує ім'я властивості `"0"`, друге - ім'я властивості `"1"` і так далі по черзі[2].

3.3 HTML - мова розмітки гіпертексту

HTML дозволяє користувачеві створювати і структурувати розділи, параграфи, заголовки, посилання і блоки для веб-сторінок і додатків.

HTML не є мовою програмування, тобто він не має можливості створювати динамічні функції. Замість цього він дозволяє організовувати і формувати документи, аналогічно Microsoft Word. При роботі з HTML використовуються прості структури коду, такі як, теги і атрибути, щоб розмітити сторінку веб-сайту.

Згідно із Довідником HTML, в даний час існує 140 тегів HTML, хоча деякі з них вже застаріли, тобто не підтримуються сучасними браузерами.

HTML-документи можна переглядати за допомогою будь-якого веб-браузера, наприклад, Google Chrome, Safari або Mozilla Firefox. Браузер читає HTML-файл і відображає його вміст, щоб користувачі інтернету могли його переглядати.

Звичайний веб-сайт включає кілька різних HTML-сторінок. Наприклад: домашні сторінки, звичайні сторінки, сторінки контактів матимуть окремі HTML-документи.

Кожна HTML-сторінка складається з набору тегів або елементів, які можна назвати будівельними блоками веб-сторінок. Вони створюють ієрархію, яка структурує контент за розділами, параграфами, заголовкам і іншим блокам контенту.

HTML-теги мають два основних типи: блок-рівень і вбудовані теги.

Елементи рівня блоку займають весь вільний простір і завжди запускають новий рядок в документі. Заголовки і параграфи - відмінний приклад блокових тегів.

Вбудовані елементи займають стільки місця, скільки їм потрібно, і не запускають новий рядок на сторінці. Вони зазвичай служать для форматування внутрішнього вмісту елементів рівня блоку. Посилання та підкреслені рядки - хороші приклади вбудованих тегів.

Через швидке зростання популярності HTML тепер вважається офіційним веб-стандартом. Найбільшим оновленням мови стало впровадження HTML5 в 2014 році[3].

HTML5

Оновлення HTML5 стало суттєвим кроком вперед у веб-дизайні, розмітці і зручність застосування. Вона надає простий спосіб роботи з графікою в браузері без звернення до додаткових модулів, наприклад до Flash, а також пропонує методи вставки аудіо та відео в веб-сторінки і згладжує деякі невідповідності.

Крім того, HTML 5 включає безліч інших удосконалень, таких як, обробка місця розташування користувача, робочі веб-процеси, керують виконанням фонових завдань, поліпшену обробку форм, доступ до пакетів локального сховища і навіть можливість перетворення веб-сторінок в веб-додатки для мобільних браузерів.

Але щодо HTML5 слід зазначити один важливий факт: оскільки ця мова перебуває у постійному розвитку, тому властивості мови різні браузери освоювали в різний час. На щастя, всі найбільш популярні новинки, що прийшли з HTML5, в кінцевому підсумку користуються підтримкою всіх основних браузерів.

3.4 CSS – каскадні таблиці стилів

CSS - мова опису зовнішнього вигляду HTML-документа. Це одна з базових технологій в сучасному інтернеті. Практично жоден сайт не обходиться без CSS, тому HTML і CSS діють в єдиній зв'язці. Каскадні таблиці стилів працюють з HTML, але CSS взаємодіє з браузером, щоб надати документу оформлення.

Мовою HTML ми створюємо розмічений текст - документ з гіперпосиланнями, таблицями, маркованими списками, різними зображеннями шрифтів, заголовками, підзаголовками і так далі. Отримуємо простір для тексту з таблицями та ілюстраціями. Інтернет винайшли вчені, і для них такий стан речей було прийнятним. Але все змінилося, коли WWW пішов в маси і свої сторінки почали створювати прості користувачі, які хотіли індивідуальності та самовираження, а також комерційні компанії зі своїми корпоративними стандартами оформлення. Загалом, веб-сторінок знадобилося індивідуальне оформлення: стиль.

У таблиці стилів ми прописуємо правила оформлення. Кожне правило складається з селектора і блоку оголошень. Селектор визначає, на які частини документа поширюється правило. Блок оголошень поміщається в фігурні дужки і складається з одного або більше оголошень, між якими ставиться крапка з комою.

Що дає використання CSS:

- Показувати один і той же документ в різних стилях.
- Декілька дизайнів сторінки для різних пристроїв.
- Зменшення часу завантаження сторінок сайту шляхом перенесення правил зображення в окремий CSS-файл. В цьому випадку браузер завантажує тільки структуру документа і дані, що зберігаються на сторінці, а стильові правила цих даних завантажуються браузером тільки один раз і кешуються.
- Простота подальшої зміни дизайну. Не потрібно правити кожную сторінку, а лише змінити CSS-файл.
- Додаткові можливості оформлення.
- Дозволяє створювати складну і пропрацьовану техніку дизайну

Сама концепція CSS підштовхувала до ідеї повністю розділити контент і оформлення документа [9].

У стародавні часи верстку документа виконували HTML-таблицями. Наприклад, якщо ви хотіли бічне меню або сайт в три колонки, то створювали в HTML таблицю з колонками відповідної ширини та прозорими кордонами. Для «роздування» окремих елементів таблиці вставлялися прозорі картинки. Загалом, було багато цікавих «милиць».

Проблема в тому, що при верстці таблицями оформлення прив'язувалася до контенту, тобто при зміні контенту часто доводилося міняти оформлення, перероблювати ці нескінченні таблиці. Це зводило з розуму дизайнерів. Вони з полегшенням зітхнули, коли з'явився CSS[5].

CSS3

Cascading Style Sheets3 (CSS3) - третє покоління стандарту CSS, яке зараз знаходиться в активній розробці. Як HTML5 для HTML, так і CSS3 для CSS став наймасштабнішою ревізією в історії стандарту.

На відміну від попередніх версій, специфікація розбита на модулі, розробка і розвиток яких йдуть незалежно. Тобто CSS3 - це теж не просто стандарт, а «набір технологій», як і HTML5. Ці технології потроху проникають в браузері. Кожна нова версія Chrome або Firefox підтримує якийсь новий шматочок CSS3, кілька нових властивостей. Тобто з кожним місяцем розширюються наші можливості по оформленню документів.

3.5 MYSQL - мова структурованих запитів

MySQL - це одна з найпопулярніших і найпоширеніших СУБД (система управління базами даних) в інтернеті. Вона не призначена для роботи з великими обсягами інформації, але її застосування ідеально для інтернет сайтів, як невеликих, так і досить великих.

MySQL відрізняється хорошою швидкістю роботи, надійністю, гнучкістю. Робота з нею, як правило, не викликає великих труднощів. Підтримка сервера MySQL автоматично включається в поставку PHP.

Важливим фактором є її безкоштовність. MySQL поширюється на умовах загальної ліцензії GNU (GPL, GNU Public License).[10]

Проектування бази даних

Перед тим як створювати базу даних, дуже важливо її вдало спроектувати, в іншому випадку, швидше за все, доведеться повертатися назад та змінювати її структуру, розбиваючи одні й об'єднуючи інші таблиці і переміщаючи різні графи з таблиць в таблицю з метою досягнення раціональних зв'язків, якими MySQL буде легше скористатися.

Для початку було б непогано сісти за стіл з аркушем паперу та олівцем і накидати добірку тих запитів, які, на ваш погляд, найчастіше будуть потрібні вам і вашим користувачам.

Наприклад, для бази даних книжкового інтернет-магазину можуть бути записані такі питання.

Скільки авторів, книг і покупців є в базі даних?

Які книги написані тим чи іншим автором?

Які книги не купувалися в цьому році?

Зрозуміло, до такої бази даних може бути зроблено і безліч інших запитів, але навіть ця добірка дасть вам уявлення про те, як слід спланувати структуру таблиць.

Наприклад, книги та номери повинні бути, скомбіновані в одній таблиці, оскільки вони тісно взаємопов'язані

Якщо взяти до уваги ці елементарні правила, то можна припустити, що для задоволення всіх цих запитів нам знадобляться як мінімум три таблиці.

Передбачається велика кількість пошуків за автором, багато з яких співпрацювали під час написання книг, а значить, будуть показані разом. Оптимальних результатів пошуку можна домогтися, якщо кожному з авторів буде дана вся інформація, що відноситься до нього. Отже, нам потрібна таблиця авторів - authors.

books (книги). Багато книг з'являються в різних виданнях. Іноді у них різні видавці, а іноді різні книги мають одне і те ж назву. зв'язки між книгами і авторами настільки складні, що для книг потрібна окрема таблиця.

customers (покупці). Причина, по якій покупці повинні знаходитися у власній таблиці, ще більш прозора - покупці можуть придбати будь-яку книгу будь-якого автора.

Реляційна база даних MySQL

Реляційні системи реалізують реляційну модель роботи з даними, яка визначає всю збережену інформацію як набір пов'язаних записів і атрибутів в таблиці.

СУБД такого типу використовують структури (таблиці) для зберігання і роботи з даними. Кожен стовпець (атрибут) містить свій тип інформації. Кожен запис в базі даних, що володіє унікальним ключем, передається в рядок таблиці, і її атрибути відображаються в стовпцях таблиці.

Нормалізація

Процес розподілу даних по таблицях і створення первинних ключів називається нормалізацією. Основна мета нормалізації - забезпечити, щоб кожна порція інформації з'являлася в базі даних тільки один раз. Тому що, дублювання даних призводить до вкрай неефективної роботи, оскільки невиправдано збільшує обсяг бази даних і уповільнює тим самим доступ до інформації. Якщо, наприклад, назви книг перераховуються і в таблиці авторів, і в таблиці книг і виникає необхідність виправити помилку в назві, потрібно буде вести пошук в обох таблицях і вносити однакові зміни всюди, де зустрічаються назви книг. Краще зберігати назви в одному місці, а в інших місцях використовувати номер.

В процесі розбиття бази даних на кілька таблиць важливо не зайти надто далеко і не створити таблиць, більше ніж потрібно, що може також привести до неефективності конструкції та уповільнення доступу до даних.

Але потрібно відкинути всі ці правила при роботі з сайтами, які мають високий рівень звернень, бо це може призвести до лишньої завантаженості бази даних. На щастя, винахідник реляційної моделі Едгар Кодд проаналізував поняття нормалізації і розбив його на три окремі схеми, названі першою, другою і третьою нормальними формами.

Нормалізація вимагає поширення даних по декількох таблицях, а це означає, що при кожному запиті буде здійснюватися кілька викликів MySQL.

Якщо на сайті, що користується високою популярністю, будуть нормалізовані таблиці та рахунок одночасно обслуговуваних користувачів піде на десятки, то швидкість доступу до бази даних істотно знизиться, тому що для їх обслуговування будуть потрібні сотні звернень до цієї бази. Якщо серйозно, то потрібно провести максимально можливу денормалізацію будь-яких даних, що часто використовуються.

Причина в тому, що дублювання даних в таблицях дозволяє істотно скоротити кількість необхідних додаткових запитів, тому що основна маса потрібних даних доступна в кожній таблиці. Це означає, що можна буде просто додати до запиту ще одну графу, і це поле стане доступно для всіх відповідних результатів, хоча доведеться змиритися з усіма згаданими раніше витратами, включаючи використання більшого обсягу дискового простору і забезпечення оновлення кожної окремої копії продубльованих даних, коли одна з них вимагає модифікації.

Звичайно, багаторазові поновлення можна комп'ютеризувати. Система MySQL надає властивість під назвою «тригери», які здійснюють автоматичні зміни бази даних відповідно до зроблених змін.

Інший спосіб надлишкових даних складається в налаштуванні PHP-програми на регулярний запуск і підтримку всіх копій в синхронізований стан [6].

3.6 Висновки до розділу 3

Для написання програмного продукту було обрано такі мови програмування: PHP, JavaScript, CSS, HTML, MySQL.

У даному розділі були наведені особливості, яких немає у аналогів, через які й були обрані дані мови програмування.

4. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

4.1 Формування бази даних

Концептуальна структура БД

Для функціонування справжнього веб-додатку потрібно створити базу даних. База даних призначена для зберігання інформації та для здійснення доступу до цих даних. Вона містить у собі зв'язані одна з одною таблиці, при цьому кожна зберігає в собі однотипну інформацію. Проектування бази даних повинно спиратися на концептуальну модель веб-додатку.

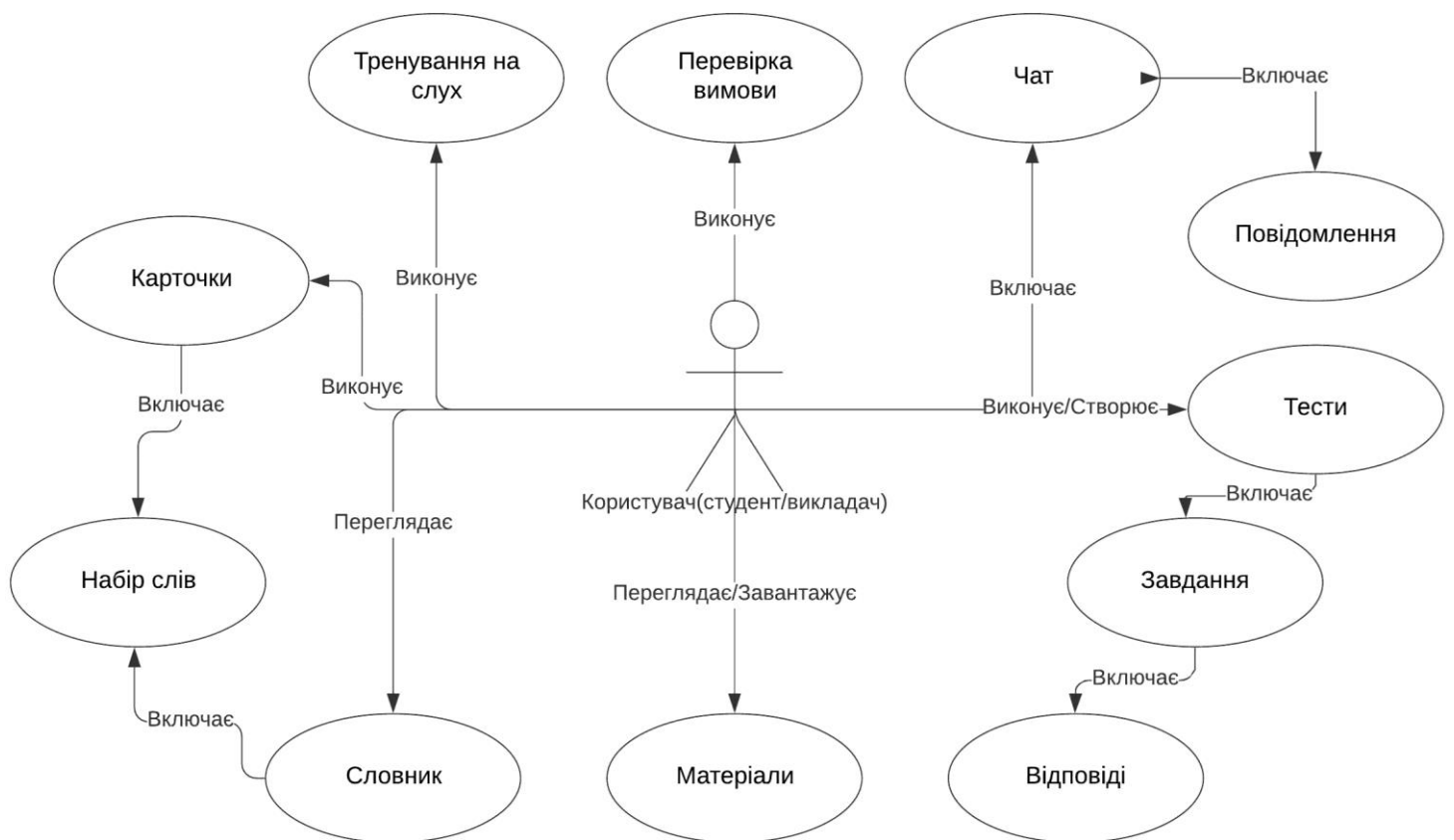


Рисунок 4.1 – Концептуальна структура БД

Фізична модель даних

Фізична модель даних, має залежати від конкретної СУБД, тобто бути відображенням системного каталогу. У фізичній моделі зберігається інформація про всі дані БД. Оскільки стандартів на об'єкти БД не існує, фізична модель залежить від конкретної реалізації СУБД. Звідси, однією і тією ж логічної моделі можуть відповідати кілька різних фізичних моделей. В логічній моделі не має значення, який саме тип даних має атрибут, але у фізичній моделі важливо описати всю інформацію про конкретних фізичних об'єктів, таких як, таблицях, колонках, індексах та інші..

В теорії було спроектовану таку модель бази даних.

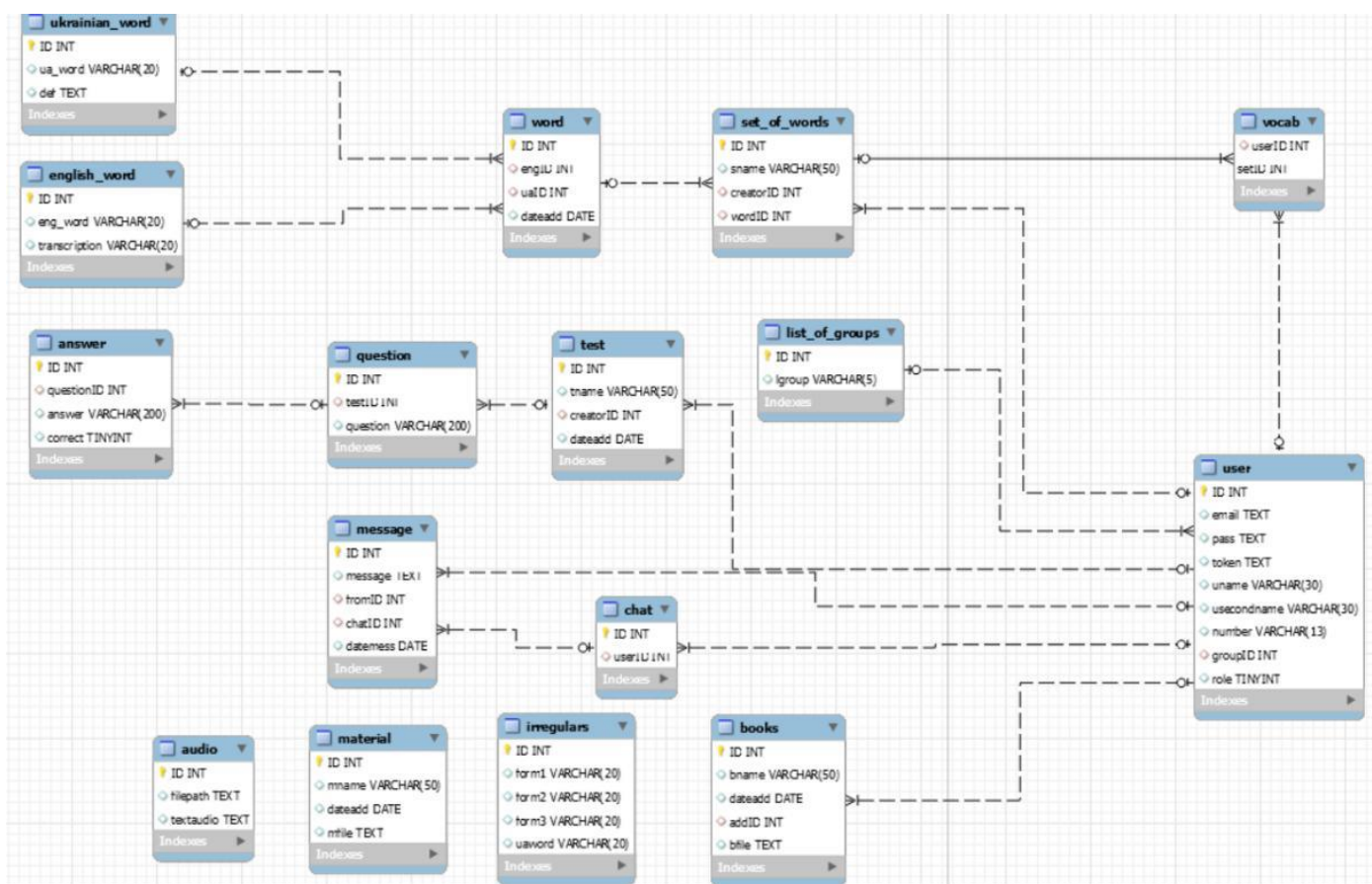


Рисунок 4.2 – Фізична модель даних (в теорії)

Проте під час практичної реалізації модель свій вигляд:

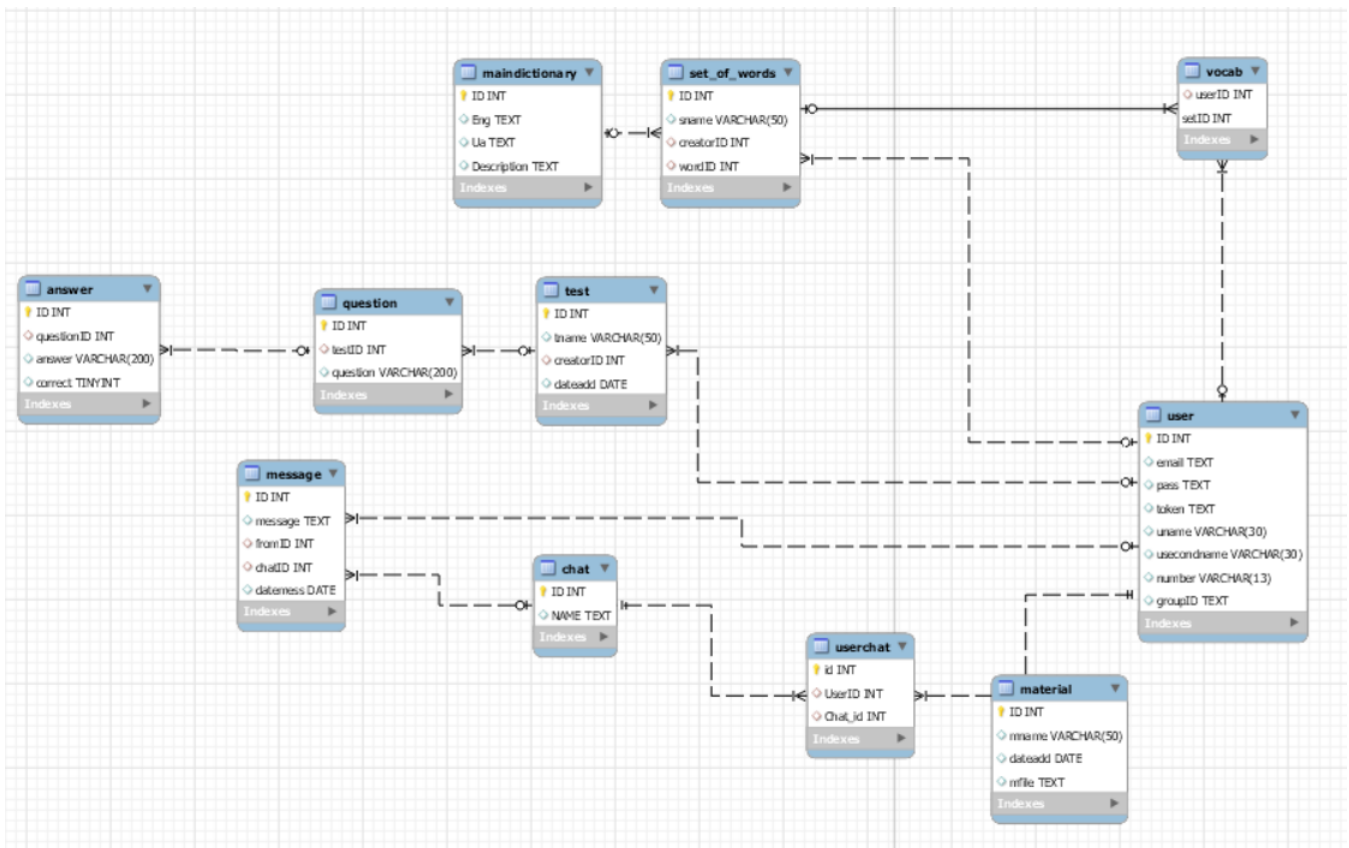


Рисунок 4.3 – Фізична модель даних (на практиці)

Структура таблиць баз даних

В таблицях, на рисунку 4.4, зберігається інформація для словника. Тобто записується ID слова, англійська і українська версія та його транскрипція. В останній таблиці записується ID користувача, яке зв'язане до ID словника цього користувача.

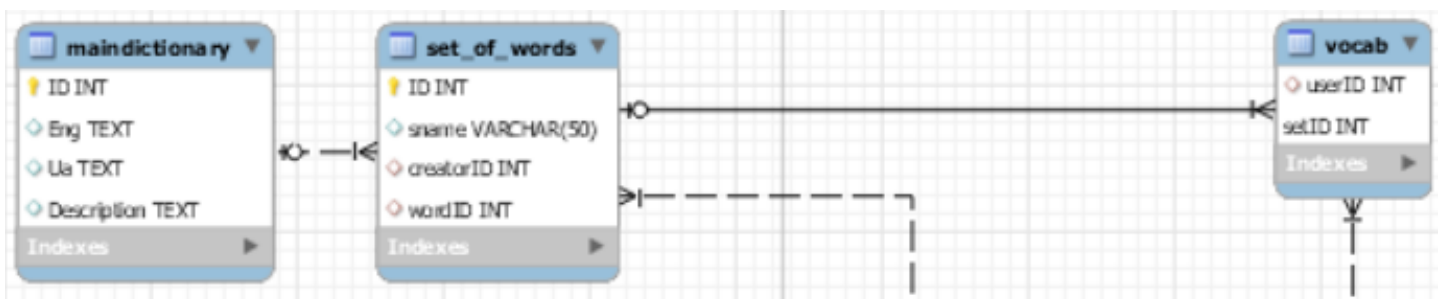


Рисунок 4.4 – Структура таблиць «Словник»

В таблицях, на рисунку 4.5, зберігається інформація про тести. Тобто записується ID тесту, що зв'язане з його творцем. Для кожного питання присвоюється власне ID, як і для кожної відповіді, та для правильної відповіді власне ID.

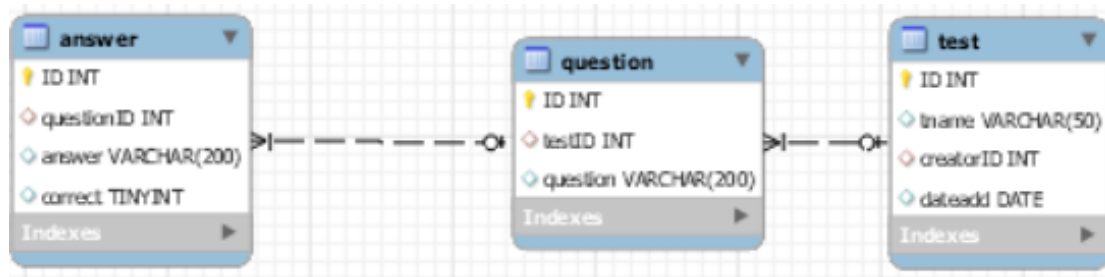


Рисунок 4.5 – Структура таблиць «Тести»

В таблицях на рисунку 4.6 зберігається інформація про повідомлення які надсилались в чат. Для кожного повідомлення присвоюється власне ID. В останній таблиці для кожного користувача присвоюється власне ID.

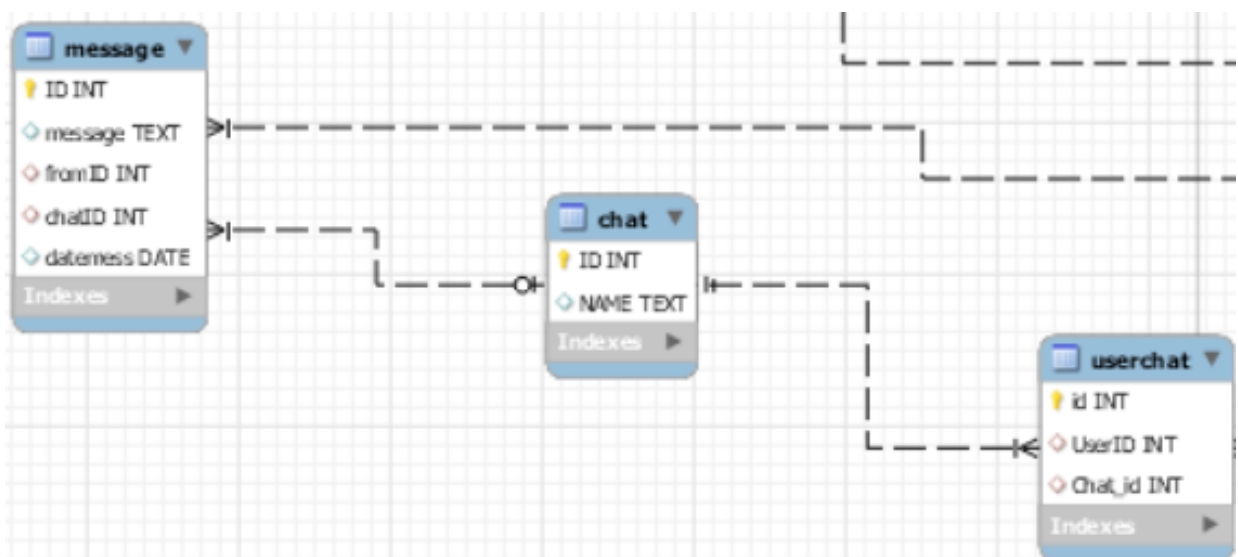


Рисунок 4.6 – Структура таблиць «Чат»

В таблиці «Матеріали», на рисунок 4.7, зберігається інформація про матеріали завантажені на сайт. Кожній завантаженій інформації присвоюється власне ID та час, коли воно було додане.



Рисунок 4.7 – Структура таблиці «Матеріали»

Приклади заповнення таблиць

Наведені нижче (рисунки 4.8 – 2.12) демонструють таблиці заповнені даними.

| id | Eng | Ua | Description |
|----|---------|------------|-------------|
| 4 | Hello | Привіт | |
| 5 | like | подобатись | |
| 6 | cat | кіт | |
| 7 | Book | Книга | |
| 8 | science | Наука | |
| 9 | table | Стіл | |
| 10 | dog | пес | |
| 11 | red | червоний | |
| 12 | green | зелений | |
| 13 | glass | скло | |
| 14 | blue | синій | |
| 15 | товари | goods | |
| 16 | House | Дом | Soon |
| 17 | hope | надіятися | Soon |
| 18 | nine | 9 | цифра |
| 19 | can | могти | |
| 20 | home | дім | дім |

Рисунок 4.8 – Приклад заповнення таблиці «Словник»

| Id | test_id | question | Answer | keyA |
|----|---------|--|------------------------------------|------------|
| 2 | 1 | ["Why", "Hellow"] | [["because", "be"], ["world"]] | ["1", "0"] |
| 3 | 2 | ["London is the capital of ..."] | [["GB", "UK", "Moscow"]] | ["2", "0"] |
| 4 | 3 | ["Qetion1?", ""] | [["1", "2", "3"]] | ["0"] |
| 5 | 4 | ["Correct answer 2", "Correct answer 3"] | [["1", "2", "3"], ["1", "2", "3"]] | ["1", "2"] |

Рисунок 4.9 – Приклад заповнення таблиці «Тест»

| ID | message | fromID | chatID | datemess |
|----|--|--------|--------|---------------------|
| 1 | Первое сообщение | 1 | 1 | NULL |
| 2 | asdasd | 1 | 1 | NULL |
| 3 | sascasdasd | 1 | 1 | 2018-12-15 01:35:59 |
| 4 | Привет людишки | 1 | 1 | 2018-12-15 06:51:18 |
| 6 | я знаю | 3 | 1 | 2018-12-15 12:56:50 |
| 7 | <script> console.log("1"); </sc... | 3 | 1 | 2018-12-15 13:12:50 |
| 9 | First message | 6 | 2 | 2018-12-16 19:25:57 |
| 10 | спропо | 6 | 2 | 2018-12-16 19:49:59 |

Рисунок 4.10 – Приклад заповнення таблиці «Чат»

| ID | email | pass | token | uname | usecondname | number | groupID | role |
|----|--------------------------|----------------------------------|------------|--------|-------------|---------------|---------|---------|
| 1 | vasia.danilyuk@gmail.com | 912ec803b2ce49e4a541068d495ab570 | ejgjWxNvbC | Vasya | Danilyk | +380966815412 | TM-62 | student |
| 4 | ssfdsfdsfd@gmail.com | 912ec803b2ce49e4a541068d495ab570 | ZX4ypjlxUX | eqwqwe | qweqwe | 423423 | TM-62 | teacher |
| 6 | yakymiv.ivan@gmail.com | 912ec803b2ce49e4a541068d495ab570 | JwVMJBbmrE | Ivan | Yakymiv | 5645468 | TM-61 | student |

Рисунок 4.11 – Приклад заповнення таблиці «Користувач»

4.2 AJAX – JS технологія

Термін AJAX був придуманий в 2005 році. Спочатку він розшифровувався як «Асинхронний JavaScript і XML», що, простіше кажучи, означало використання набору методів, вбудованих в JavaScript, для обміну даними між браузером і сервером у фоновому режимі. Чудовим прикладом застосування цієї технології є Google Maps де нова ділянка карти завантажується з сервера за потреби, на що не потребується оновлення всієї сторінки.

Використання AJAX не тільки призводить до істотного зниження обсягу обміну даними, а й забезпечує плавну динамічність веб-сторінок, роблячи їх поведінку характерним для самостійних додатків. В результаті значно поліпшується користувацький інтерфейс і прискорюється реакція на дії користувача.

Що таке AJAX

Сучасний AJAX почався в 1999 році з випуску Internet Explorer 5, де був переставлений новий ActiveX-об'єкт для XMLHttpRequest. Розроблена в корпорації Microsoft технологія ActiveX передбачає використання додаткових програмних модулів, що встановлюються на комп'ютер. Пізніше розробники інших браузерів підтримали це нововведення, але замість застосування ActiveX вони розробили функціональний модуль, який став невід'ємною частиною інтерпретатора JavaScript.

Але на той час вже з'явилася рання форма AJAX, яка використовує на сторінці приховані фрейми, які взаємодіють з сервером у фоновому режимі. Найпершими споживачами цієї технології були учасники форумів, які залучили її для опитування та показ нових повідомлень без перезавантажень сторінки.

Переваги AJAX:

- Можливість створення зручного Web-інтерфейсу
- Активна взаємодія з користувачем
- Часткове перезавантаження сторінки, замість повної
- Зручність використання

Що можна зробити з допомогою AJAX

- Елементи інтерфейсу

В першу чергу AJAX корисний для форм і кнопок, пов'язаних з елементарними діями: додати в кошик, підписатися, та інші.

Зараз - в порядку речей, що такі дії на сайтах здійснюються без перезавантаження сторінки.

- Динамічне підвантаження даних

Наприклад, дерево, яке при розкритті вузла запрошує дані у сервера.

- Живий пошук

Живий пошук - класичний приклад використання AJAX, взятий на озброєння сучасними пошуковими системами. Користувач починає друкувати пошукову фразу, а JavaScript пропонує можливі варіанти, отримуючи список найімовірніших доповнень з сервера[4].

4.3 jQuery – бібліотека JS

При всій гнучкості та ефективності JavaScript, а також при всьому достатку наявних в цій мові вбудованих функцій все ж зберігається потреба в додаткових рівнях коду, що дозволяють спростити, наприклад, отримання ефектів анімації, обробку подій і застосування технології Ajax, тобто зробити те, чого не можна досягти застосуванням звичайних засобів JavaScript або CSS. В результаті постійної

війни браузерів, гарантувати однаковий зовнішній вигляд веб-сторінок на всіх пристроях часом не можна було. Проте можна застосовувати код JavaScript, що враховує всі розбіжності всієї лінійки браузерів і їх версій, випущених за останні роки.

На даний час є відчуття, що вже не обійтись без таких заклинань, оскільки Internet Explorer компанії Microsoft в багатьох областях наздогнав стандарти, в браузері Opera в основу вирішено було застосовувати Webkit (ту ж саму технологію, яка використовувалася компанією Google), а компанія Apple пішла з ринку браузерів для PC-сумісних комп'ютерів.

І все ж успадкована несумісність ще залишилася, а багато хто до цього часу користуються старими браузерами. Плюс до того, якщо є бажання створити щонебудь більш вражаюче, ніж звичайні спецефекти, вам все ще доведеться писати досить істотний обсяг коду на JavaScript.

Для заповнення прогалин було розроблено безліч бібліотек функцій, призначених для зведення до мінімуму різниці між браузерами, і деякі з цих бібліотек дозволяють, легко прив'язуватися до об'єктної моделі документа – DOM, що полегшують роботу з Аjax, подіями і анімацією.

Я вибрав найбільш широко використовувану бібліотеку jQuery, яка сьогодні, встановлена більш ніж на 60% сайтів і має попит ширший, ніж всі її конкуренти, разом узяті.

Використовуючи jQuery, можна отримати не тільки кросбраузерну сумісність вельми високого рівня, включаючи навіть сумісність з Internet Explorer, але і швидкий і легкий доступ до операцій з HTML і DOM, можливість використання спеціальних функцій для безпосередньої роботи з CSS, управління подіями, потужні засоби для створення професійних ефектів і анімації, а також функції для управління обміном даними з сервером за технологією Аjax. Крім того, jQuery є основою для широкого кола додаткових модулів та інших допоміжних програм.

Використання jQuery ніхто не нав'язує, і деякі борці за чистоту мови програмування ніколи не використовують бібліотеку, вважаючи за краще створювати власний спеціалізований набір функцій.

Використання мережі доставки контенту

Бібліотека jQuery підтримується декількома мережами доставки контенту (CDN). Якщо користуватися однією з них, то можете позбавити себе від турбот, пов'язаних із завантаженням нових версій, і викладати їх на сервер, просто вказавши прямі посилання, які підтримуються цими мережами. До всього іншого, ці мережі надають свої послуги абсолютно безкоштовно і зазвичай використовують канали з високою пропускнуою здатністю, які є найшвидкіснішими на світі. Крім того, CDN-мережі зазвичай зберігають свій контент в декількох різних географічних пунктах і надають пріоритет файлу з найближчого до користувача сервера, гарантуючи тим самим найбільш швидку доставку.

Загалом, якщо не потрібно вносити зміни до початкового коду jQuery (для чого потрібно його розміщення на ваших власних веб-серверах) і у ваших користувачів гарантовано є живе інтернет-з'єднання, то, швидше за все, найкращим варіантом буде використання CDN-мереж. Тим більше що користуватися ними досить просто. Достатньо знати ім'я потрібного файлу і використаного для його завантаження кореневого каталогу CDN.

Синтаксис jQuery

Найбільше людей, раніше не знайомих із jQuery, дивує символ \$, який діє як фабричний метод jQuery. Він був вибраний з розрахунку допустимості в JavaScript, стислості та відмінності від імен звичайної змінної, об'єкта чи функції. Ним позначається виклик функції jQuery. Задум його використання полягає в збереженні стислості та приємного зовнішнього вигляду коду, а також позбавлення від зайвої набору тексту при кожному зверненні до jQuery. Крім того, при вигляді цього символу інші, раніше незнайомі з вашим кодом розробники відразу ж розуміють, що в коді використовується jQuery.

Спеціальні ефекти

У чому дійсно досягла успіху бібліотека jQuery, так це в створенні спецефектів. Можна, звичайно, скористатися переходами CSS3, але їх динамічне управління з JavaScript не буде настільки ж простим, а з використанням jQuery все зведеться до простого вибору одного або декількох елементів з застосуванням до них одного або декількох ефектів. Основними доступними ефектами є зникнення і поява, прояв і розчинення, ковзання, а також анімації, які можуть використовуватися по одному, все разом узгоджено за часом або один за одним. Підтримуються також зворотні виклики, що являють собою функції або методи, які викликаються тільки один раз після закінчення операції. У наступному розділі перераховується ряд найбільш корисних jQuery-ефектів, кожним з яких підтримуються три аргументи.

Відсутність аргументів. Коли аргументи не надаються, метод викликається негайно і не потрапляє в чергу анімації.

- Тривалість.

Коли надається це значення, ефект буде спостерігатися протягом призначеного часу, яке може бути задано в мілісекундах або ж рядками `fast` (швидко) або `slow` (повільно).

- Зміна швидкості виконання ефекту.

У бібліотеці jQuery тільки два варіанти зміни швидкості: `swing` (з прискоренням) і `linear` (лінійне зсування). За замовчуванням використовується варіант `swing`, який задає більш природна зміна швидкості ефекту, ніж `linear`.

- Функція зворотного виклику.

Якщо надана функція зворотного виклику, вона буде викликана відразу ж після завершення роботи методу створення ефекту.

4.4 Запити до бази даних MySQL за допомогою PHP

Сенс використання PHP в якості інтерфейсу до MySQL полягає в форматуванні результатів SQL-запит і надання їм зовнішнього вигляду, призначений для виведення

на веб-сторінку. Володіючи можливістю входу в систему MySQL за допомогою свого імені користувача та пароля, можна зробити те ж саме і з PHP. Але замість використання командного рядка MySQL для введення команд і перегляду вихідної інформації потрібно буде створити рядки запиту, а потім передати їх MySQL.

Відповідь MySQL надійде у вигляді структури даних, яку PHP зможе розпізнати, а не у вигляді того форматowanego екранного виведення, який спостерігався в процесі роботи з командним рядком. Потім за допомогою команд PHP можна буде витягати дані і приводить їх до формату веб-сторінки.

Процес використання MySQL за допомогою PHP полягає в наступному:

1. Підключення до MySQL і вибір бази даних, яка буде використовуватися.
2. Створення рядка запиту.
3. Виконання запиту.
4. Витяг результатів і висновки їх на веб-сторінку.
5. Повторення кроків з 2-го по 4-й доти, поки не будуть витягнуті всі необхідні дані.
6. Відключення від MySQL

Перед початком роботи з даними всередині MySQL, потрібно відкрити з'єднання з сервером СУБД.

У PHP це робиться за допомогою стандартної функції `mysqli_connect ()`. Функція повертає результат - ресурс з'єднання. Даний ресурс використовується для всіх наступних операцій з MySQL.

Але щоб виконати з'єднання з сервером, необхідно знати як мінімум три параметри: Адреса сервера СУБД, Логін, Пароль.

4.5 Висновки до розділу 4

У розділі 4 були наведені інструменти, що використовувались для програмної реалізації програмного продукту.

Інструменти були вибрані саме для тих програм, що наведені в розділі 3.

5. РОБОТА КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ

5.1 Системні вимоги

Для забезпечення коректної та безвідмовної роботи інформаційної системи ефективного моніторингу хронічних захворювань персональний комп'ютер повинен мати процесор не гірше, ніж

- Intel ®
- Pentium ®
- Celeron ®
- Xeon™

Тактовою частотою не менше 1,8 GHz AMD 6.

Також комп'ютеру користувача повинно бути доступно не менше 2 Gb оперативної пам'яті та графічне ядро не гірше, ніж Intel ® HD Graphics 2000, що еквівалентно графічним картам з об'ємом пам'яті не менше, ніж 128 Mb. Для запуску програмного модуля необхідно встановити Open Server.

Веб-додаток повинен забезпечувати коректну роботу користувачів в наступних версіях інтернет-браузерів:

- Internet Explorer 10+,
- Mozilla Firefox 38+,
- Safari 6+,
- Chrome 38+,
- Opera 28+;

Веб-додаток може не підтримувати частину функціоналу в інтернет-браузерах більше ранніх версій (для перерахованих вище інтернет-браузерів), а також інтернет-браузери інших розробників;

В інтернет-браузері повинна бути включена підтримка JavaScript та Cookies.

5.2 Сценарій роботи користувача від імені студента

1 Форма Реєстрації\Авторизації (Рисунок 5.1)

Початкова сторінка містить дві форми Register та Sign Up, де користувач може ввести свою особисту інформацію. Для нових користувачів форма реєстрації містить поля, а саме Email та Password, Name та Second name, Phone, Group та поле вибору Role являє собою список, що розкривається, та містить у собі два значення student або teacher. Після реєстрації користувача перекине на форму авторизації, де він може ввести Email та Password, для входу в систему.

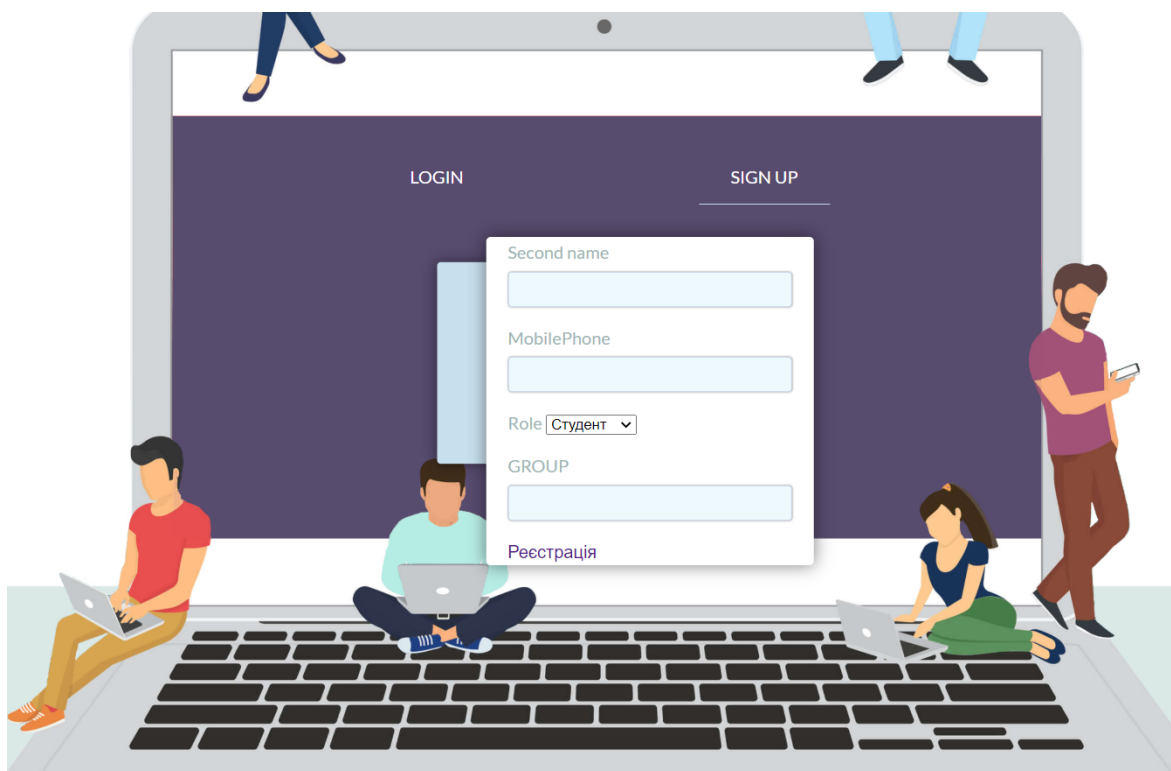


Рисунок 5.1 – Реєстрації\Авторизація

2 Розділ Головна сторінка (рисунок 5.2) містить такі можливості:

- Перегляд інформації свого акаунту.
- Перегляд статистики та кількості набраних балів за пройдені тести
- Вибір головних розділів сайту, таких як Словник, Вправи та Тести
- Крім головних розділів, також є можливість вибрати всі інші можливості сайту, такі як Чат, вибір Вправ, та Матеріалів.
- Також в користувача є можливість вийти зі свого акаунту та редагувати інформацію свого акаунту.

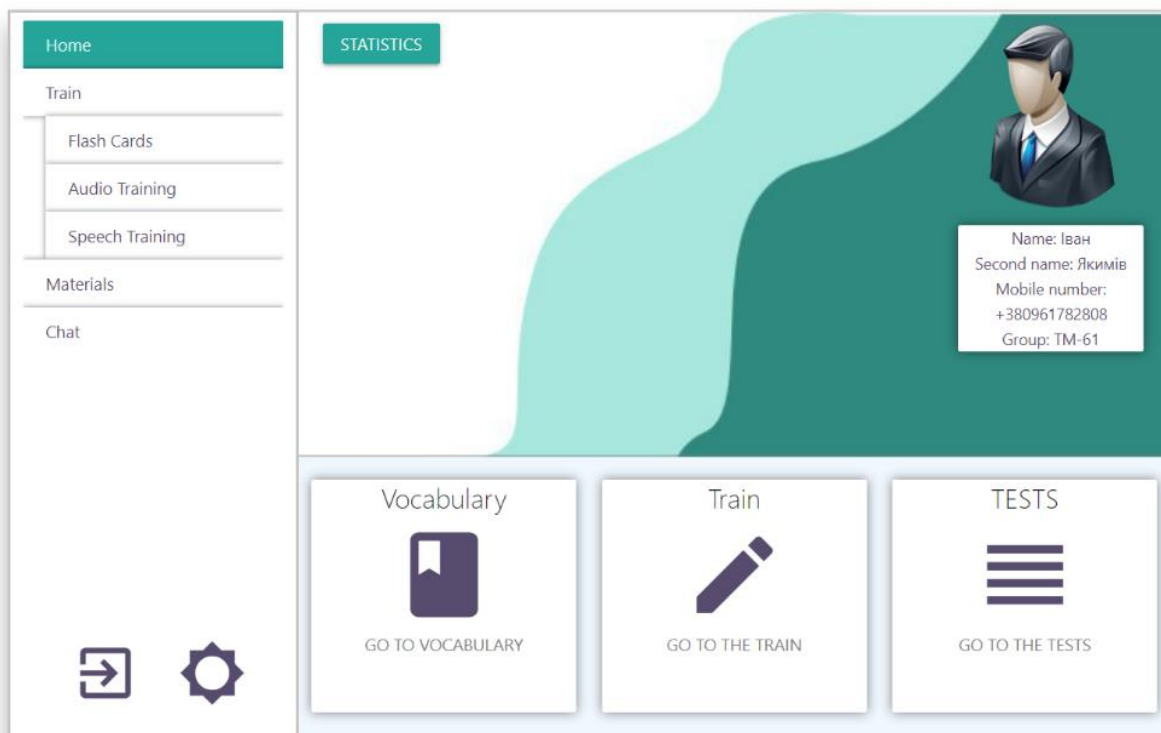


Рисунок 5.2 – Головна сторінка

3 Розділ **Тести** (Рисунок 5.3) містить такі можливості:

- Перегляд всіх тестів, сортування відбувається за датою додавання.
- Після вибору тесту, студенту надається можливість виконати тест та отримати за нього певну кількість балів.

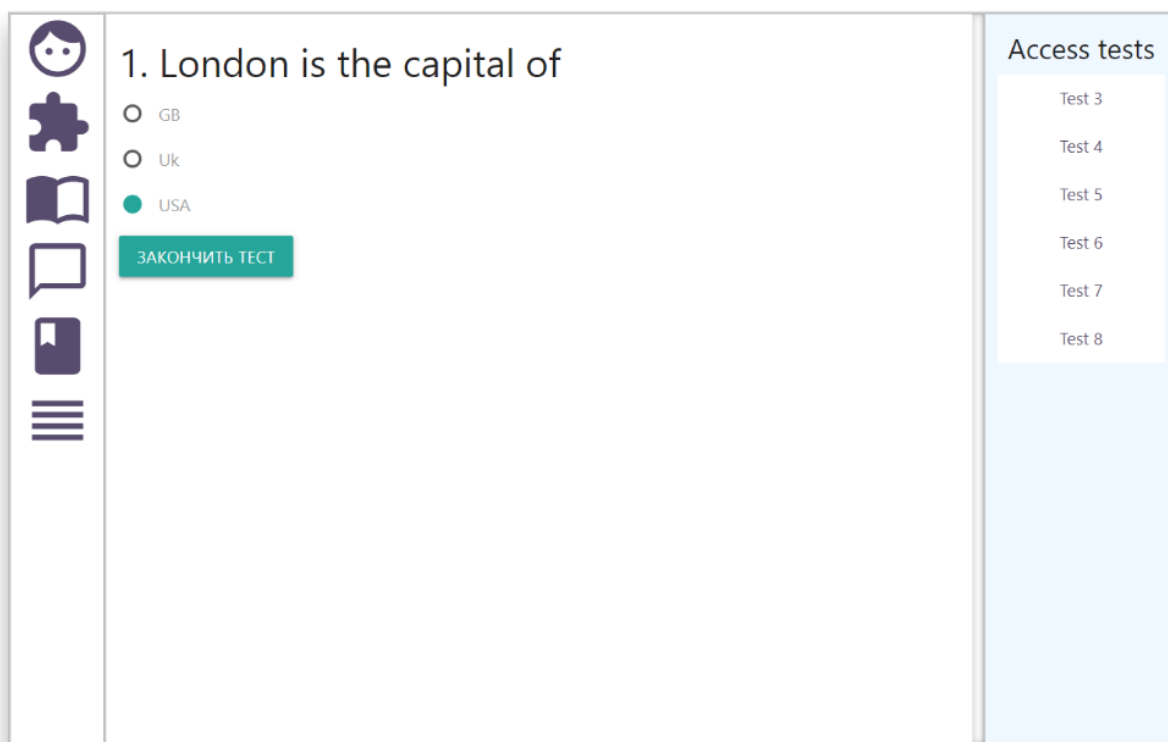
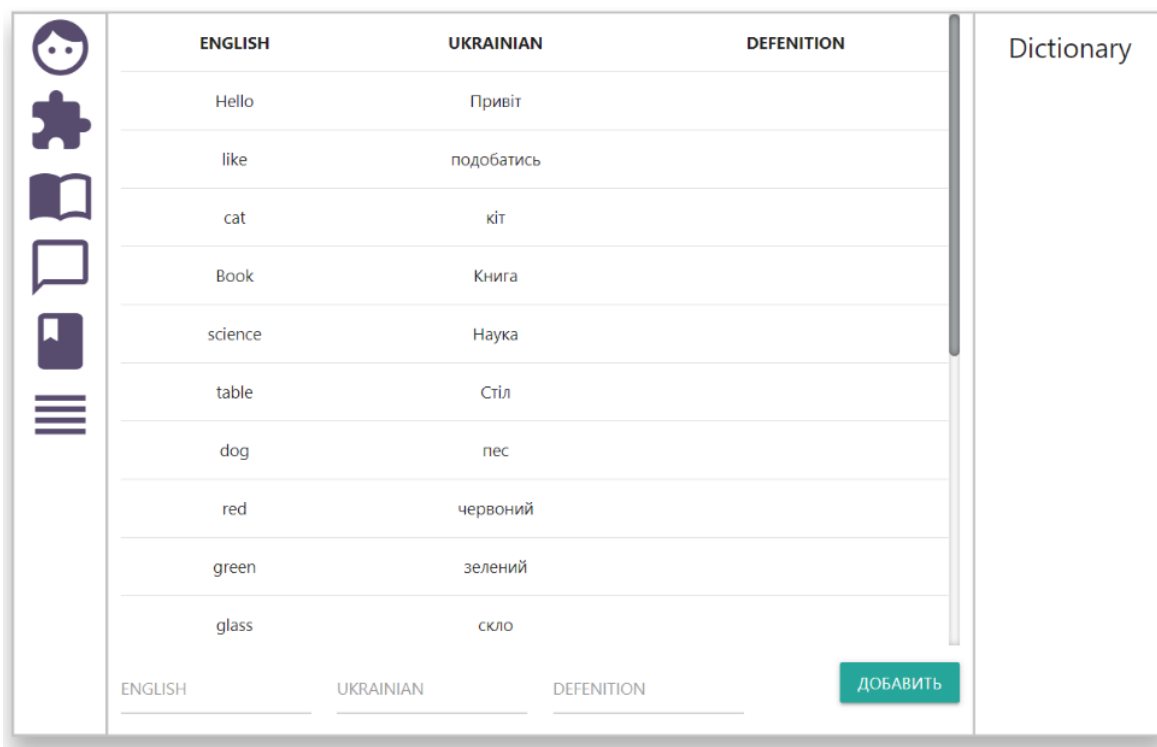


Рисунок 5.3 – Тест

4 Розділ **СЛОВНИК** (Рисунок 5.4) містить такі можливості:

- Перегляд всіх слів у словнику
- Пошук слова у словнику
- Перегляд всіх створених наборів слів
- Додавання нового слова, після заповнення полів.



| ENGLISH | UKRAINIAN | DEFENITION |
|---------|------------|------------|
| Hello | Привіт | |
| like | подобатись | |
| cat | кіт | |
| Book | Книга | |
| science | Наука | |
| table | Стіл | |
| dog | пес | |
| red | червоний | |
| green | зелений | |
| glass | скло | |

ENGLISH UKRAINIAN DEFENITION

ДОБАВИТЬ

Рисунок 5.4 – Словник

5 Розділ **Вправи** (Рисунок 5.5 містить такі підрозділи, як

- «**Flash Cards**» (Рисунок 5.6) : перед користувачем набір карточок, необхідно знайти відповідності. При натиску на карточку вона стає активною, очікується вибір наступної карточки користувача. При виборі наступної карточки, виконується перевірка на відповідність.
- «**Тренування на слух**» (Рисунок 5.7): користувач прослуховує аудіо довжиною в 1-2 хвилини або пісню, також його доступний повний текст запису (тільки під час прослуховування), далі він виконує вправу, вставити пропуски в тексті,
- «**Перевірка вимови**» (Рисунок 5.8) : користувач має можливість записувати власну вимову та після чого може прослуховувати та перевіряти, чи правильно він сказав те чи інше. Користувачу може надаватись готовий текст для читання та вимови, та записаний його викладачем правильний приклад вимови цього тексту.

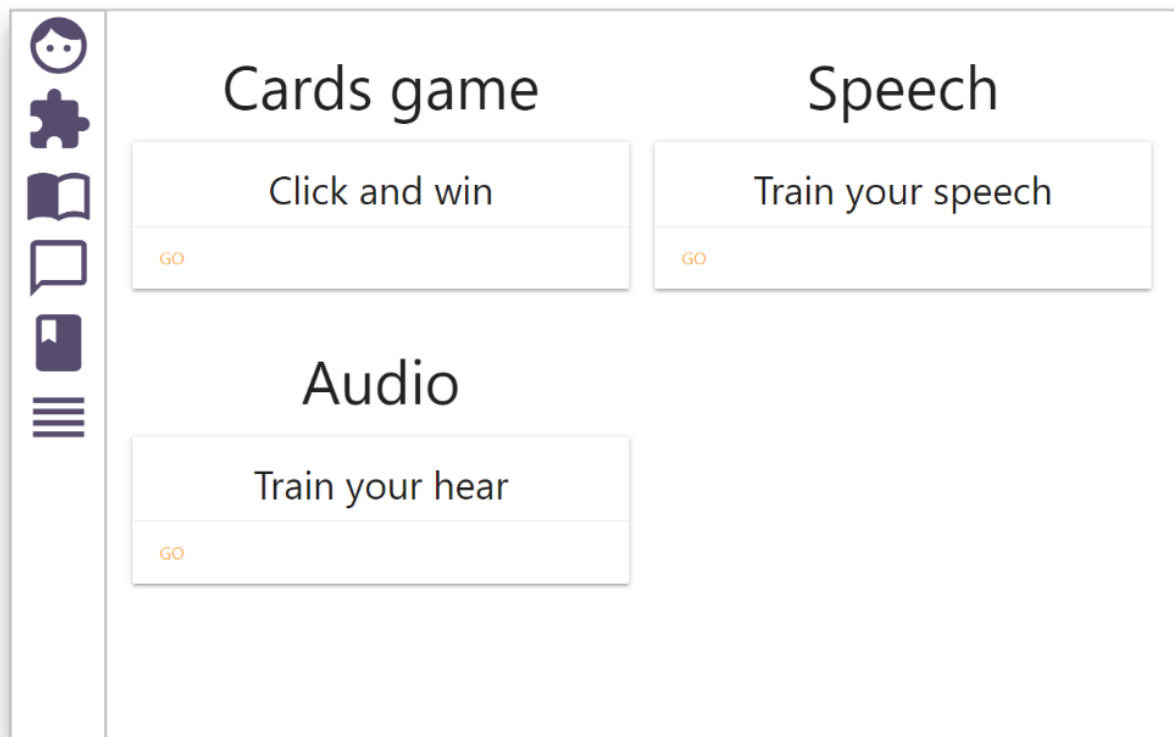
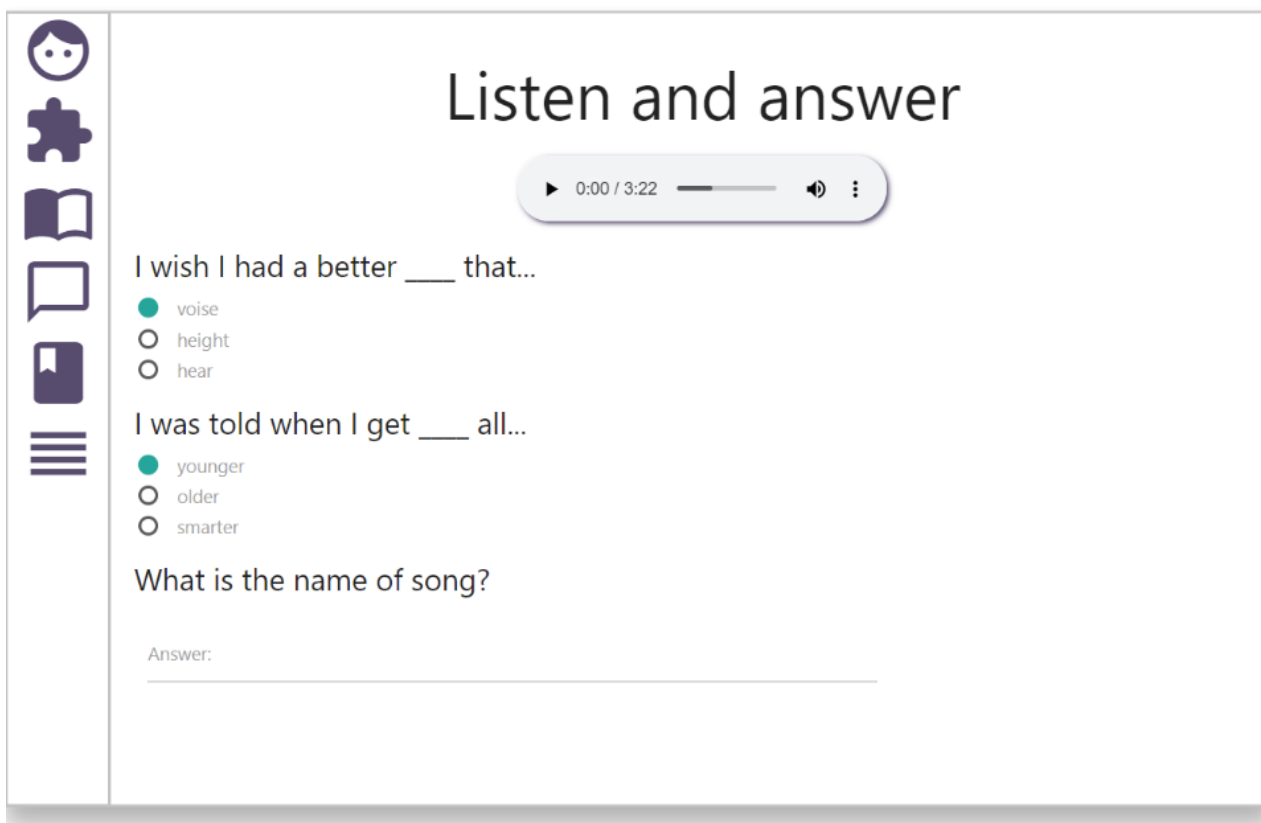


Рисунок 5.5 – **Вправи**



Рисунок 5.6 – **Flash Cards**



The interface features a vertical sidebar on the left with icons for a user profile, puzzle, book, speech bubble, document, and a menu. The main area is titled "Listen and answer". Below the title is a video player with a progress bar at 0:00 / 3:22 and a volume icon. The exercise consists of three questions, each with a list of options:

I wish I had a better ____ that...

- ☒ voice
- ☐ height
- ☐ hear

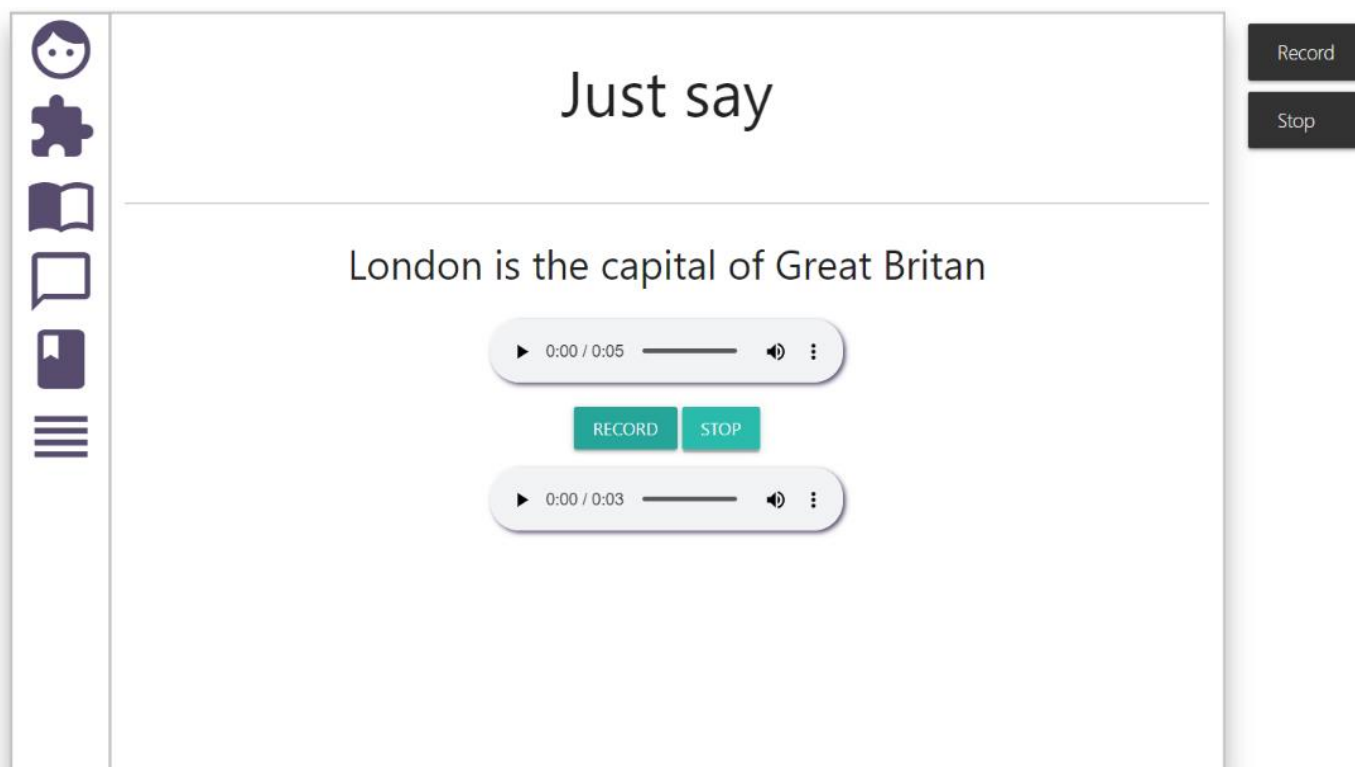
I was told when I get ____ all...

- ☒ younger
- ☐ older
- ☐ smarter

What is the name of song?

Answer: _____

Рисунок 5.7 –Тренування на слух



The interface features a vertical sidebar on the left with icons for a user profile, puzzle, book, speech bubble, document, and a menu. The main area is titled "Just say". Below the title is a text prompt: "London is the capital of Great Britan". Below the prompt are two video player controls. The first control has a progress bar at 0:00 / 0:05 and a volume icon. Below it are two buttons: "RECORD" and "STOP". The second control has a progress bar at 0:00 / 0:03 and a volume icon. On the right side of the interface, there are two buttons: "Record" and "Stop".

Рисунок 5.7 – Перевірка вимови

6 Розділ Комунікації-Чат (Рисунок 5.9) містить такі можливості:

- Перегляд чату своєї групи.
- Відправляти повідомлення в чат, де його побачать інші користувачі цього чату
- При натиску на користувача з чату відкривається його профіль, де можна почати з ним особистий чат шляхом натиску на кнопку «Send a message».

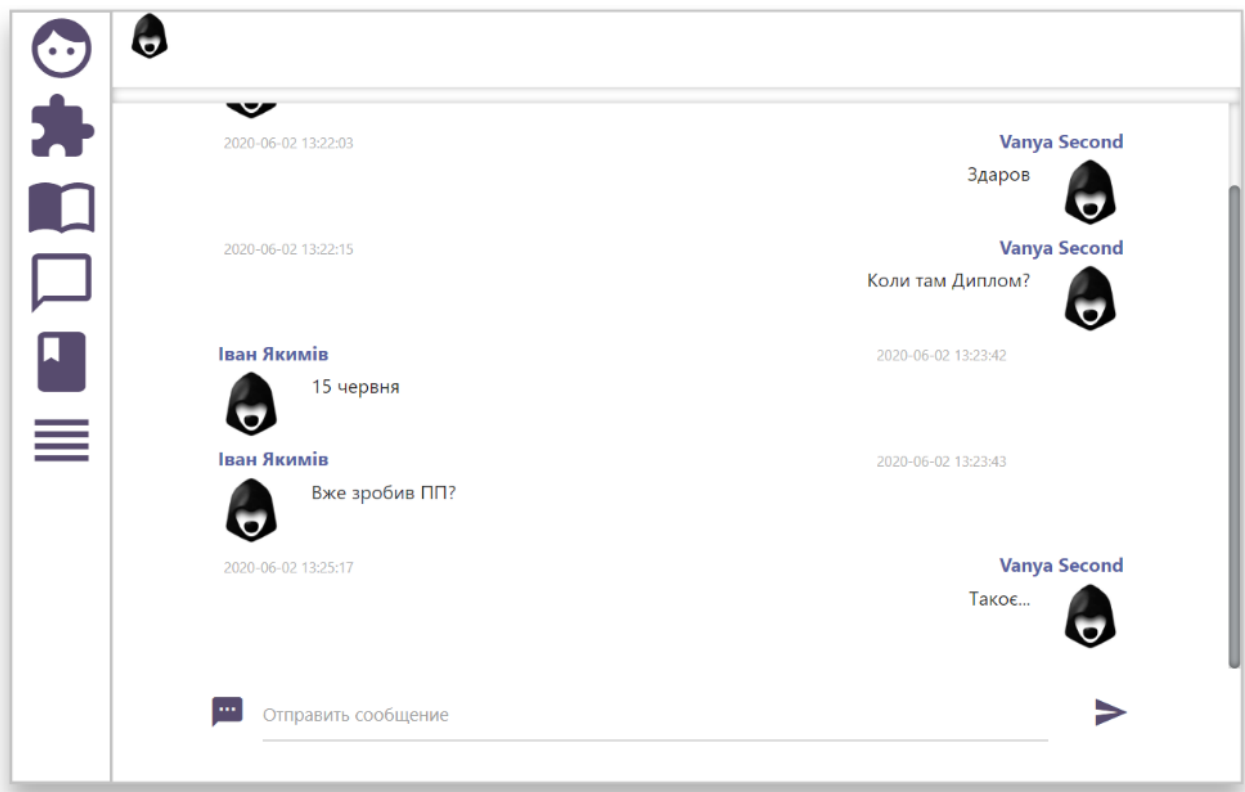


Рисунок 5.9 – Чат

7 Розділ Комунікації-Чат містить такі можливості:

- «**Книжки**» (Рисунок 5.10) користувачу надається можливість перегляду книжок, які завантажив викладач, в ПДФ форматі.
- «**Відео**» (Рисунок 5.11) користувачу надається можливість перегляду відео з YouTube, які завантажив викладач.



Рисунок 5.10 – Книжки

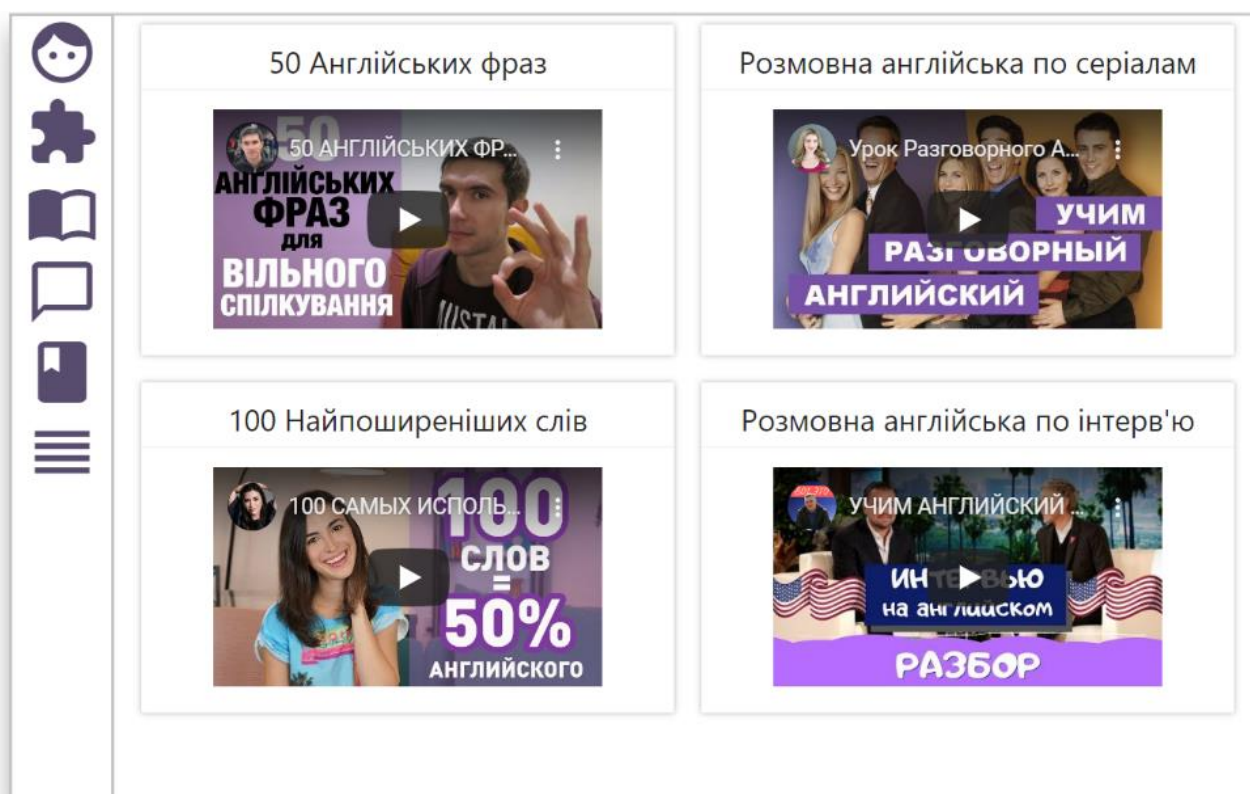


Рисунок 5.11 – Відео

5.3 Сценарій роботи користувача від імені викладача

Для користувача, що зареєструвався від імені викладача доступні всі ті самі функції, що і студенту. Проте існують додаткові можливості, такі як, перегляд успішності студентів та створення нових тестів.

8 Форма для створення нових тестів (Рисунок 5.12) містить такі можливості:

- Введення питання для тесту
- Додання варіантів відповіді на питання
- Вибір правильної відповіді
- Збереження тесту для подальшого проходження студентами

Рисунок 5.12 – Форма для створення тестів

5.4 Висновок до розділу 5

У даному розділі були представлені системні вимоги для користування додатку, інструкції, щодо роботи із веб-додатком для вивчення англійської мови.

Продемонстровано можливі сценарії роботи користувача з системою, показано зручність інтерфейсу користувача.

ВИСНОВОК

Для розв'язання проблем, пов'язаних із вивченням англійської мови у вищих навчальних закладах, було проведено дослідження. Під час дослідження було розглянуто актуальність навчання іноземних мов у сучасному світі, були проаналізовані найрізноманітніші способи вивчення та підсумування їхніх плюсів та мінусів, після чого було вирішено розробляти веб-додаток.

Завдяки аналізу найпопулярніших способів навчання іноземних мов було сформоване дерево проблем та дерево цілей, що допомогло сформулювати основні вимоги до інтерфейсу та функціональні вимоги до кожного з розділів.

Програму можна використовувати для вивчення англійської мови не тільки у вищих навчальних закладах, а у будь-яких організаціях де є студенти та викладачі, тобто школах, коледжах, тренінгах та курсах.

Для роботи з даним програмним забезпеченням необхідний лише комп'ютер середньої потужності та нові версії браузерів, що можна безплатно завантажити з офіційних сайтів.

Користувач має змогу вивчати англійську мову з будь-якого місця розташування, де є інтернет. Для цього у нього є декілька способів, таких як вправи на перевірку вимови, гра для вивчення нових слів та тести, які у свою чергу, може створювати та перевіряти викладач. Також студент має змогу листування зі своїми одногрупниками та ділитися з ними своєю успішністю.

Виконуючи цей проект я спробував використати можливість вивчення мови під контролем викладача, який знає рівень кожного студента та може контролювати прогрес у вивченні мови. Об'єднавши цю можливість з базовим функціоналом всіх способів навчання, я отримав веб-додаток, який буде абсолютно безкоштовним та має увесь необхідний функціонал щоб максимально ефективно вивчати мову під час здобуття вищої освіти.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Робин Никсон — “Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5” — 68с.— Навчальний посібник. — 2016.
2. Дуглас Крокфорд — “JavaScript: сильные стороны”— 27с.— Навчальний посібник. — 2012.
3. Бен Хеник— “HTML и CSS. Путь к совершенству”— 32с.— Навчальний посібник. — 2014.
4. Шелли Пауэрс— “Добавляем Ajax”— 12с.— Навчальний посібник. — 2015.
5. Бен Хеник— “HTML и CSS. Путь к совершенству”— 47с.— Навчальний посібник. — 2014.
6. Робин Никсон — “Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5” — 193с.— Навчальний посібник. — 2016.
7. “Чекбоксы - це що таке?” [Електронний ресурс] – 2019 – Режим доступу до ресурсу: <https://ogimp.ru/uk>
8. “Знайомство з PHP” [Електронний ресурс] – 2013 – Режим доступу до ресурсу: <https://wiki.cuspu.edu.ua/index.php>
9. “Що таке CSS” [Електронний ресурс] – 2017 – Режим доступу до ресурсу:https://css.in.ua/article/shcho-take-html_10
10. “Web-технології та Web-дизайн” [Електронний ресурс] – 2015 – Режим доступу до ресурсу:<https://dl.sumdu.edu.ua/textbooks/86975/413008/index.html>

ДОДАТОК 1

Веб-додаток для вивчення англійської мови

Специфікація

УКР.НТУУ"КПІ ім. Ігоря Сікорського" _ТЕФ_АПЕПС_ ТМ61162_20Б

Аркушів 2

Київ – 2020

| Позначення | Найменування | Примітки |
|---|---|---|
| Документація | | |
| УКР.НТУУ«КПІ ім. Ігоря Сікорського»_ТЕФ_АПЕ ПС_ТМ61162_20Б | Якимів Записка.docx | Пояснювальна записка |
| Компоненти | | |
| УКР.НТУУ«КПІ ім. Ігоря Сікорського»_ТЕФ_АПЕ ПС_ТМ61 | www.POSTAVTE100. com\index.php | Модуль авторизації\реєстрації |
| УКР.НТУУ«КПІ ім. Ігоря Сікорського»_ТЕФ_АПЕ ПС_ТМ61162_20Б | www.POSTAVTE100. com\example.php www.POSTAVTE100. com\test.php www.POSTAVTE100. com\traine.php www.POSTAVTE100. com\material.php www.POSTAVTE100. com\dictionary.php | Модулі додатку |
| УКР.НТУУ«КПІ ім. Ігоря Сікорського»_ТЕФ_АПЕ ПС_ТМ61162_20Б | www.POSTAVTE100. com\img | Матеріали для відображення в інтерфейсі |

ДОДАТОК 2

Веб-додаток для вивчення англійської мови

Текст програми

УКР.НТУУ «КПІ ім. Ігоря Сікорського»_ТЕФ_АПЕПС_ТМ61162_20Б

Аркушів 9

Київ – 2020

```

<!--Опис початкової сторінки сайту -->
<!--Починаємо сесію -->
<?php
session_start(); <!--Починаємо сесію -->
if(isset($_SESSION['token']))
{
    if($_SESSION['token']==1)
        header ('Location: example.php');
}
?>

<html>

<head>
<!--Задаємо розміщення стилей та іконок -->
    <meta charset="UTF-8">
    <link rel="shortcut icon" type="image/x-icon"
href="https://static.codepen.io/assets/favicon/favicon-
8ea04875e70c4b0bb41da869e81236e54394d63638a1ef12fa558a4a835f1164.ico" />
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons"
rel="stylesheet">
    <!--Import materialize.css-->
<!--Підключаємо потрібні файли -->
    <script src="js/jquery-3.3.1.min.js"></script>
    <!--Let browser know website is optimized for mobile-->
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta charset="utf-8">
    <link rel="mask-icon" type=""
href="https://static.codepen.io/assets/favicon/logo-pin-
f2d2b6d2c61838f7e76325261b7195c27224080bc099486ddd6dccb469b8e8e6.svg"
color="#111" />
    <title>Sign Up</title>
    <link href='https://fonts.googleapis.com/css?family=Lato'
rel='stylesheet' type='text/css'>

    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/meyer-reset/2.0/reset.min.css">
<!-- Задаємо стилі інтерфейсу -->
    <style>
        * {
            box-sizing: border-box;
        }
    html{
        background-image: url(../img/02.png);
        background-size: cover;
        position: unset;
        background-position: center;
        /* background-position-y: center; */
        height: 100%;
    }

```

```

    }
    .panel__box .active{
        height: 230px;
    }
.view__more {
    position: absolute;
    left: 0;
    top: 0;
    padding: 20px 40px;
    background: #616fa5;
    color: #fff;
    transition: all .3s;
}
.view__more:hover {
    background: #717eae;
}

body {

    font-family: 'Lato', sans-serif;
}

a {
    text-decoration: none;
}
#signInBox
    {
        height: 230px;
    }
.panel {

    width: 600px;
    margin: auto;

}
.panel__menu {
    width: 100%;
    float: left;
    margin: 20px 0 30px;
    margin-top: 23%;
    position: relative;
}
    border: none;
    background: #C8DFED;
    height: 1px;
    margin: 0;
    transition: all 0.5s;
}
.panel__box input[type="email"], .panel__box input[type="password"] {

```

```

margin-top: 10px;
width: 100%;
float: left;
background: #EEF9FE;
border: 1px solid #CDDBEF;
border-radius: 3px;
padding: 7px 10px;
}
.panel__box input[type="submit"] {
float: right;
cursor: pointer;
border: none;
padding: 11px 40px;
background: #A5E434;
border-radius: 30px;
color: #fff;
}

</style>

<script>
    window.console = window.console || function(t) {};
</script>

<script>
    if (document.location.search.match(/type=embed/gi)) {
        window.parent.postMessage("resize", "*");
    }
</script>

</head>

<body translate="no">

    <div class="panel">
        <ul class="panel__menu" id="menu">
            <hr />
            <li id="signIn"> <a href="#">Login</a></li>
            <li id="signUp"><a href="#">Sign up</a></li>
        </ul>
        <!--Задаємо поля вводу для аторизаціх -->
        <div class="panel__wrap">
            <div class="panel__box active" id="signInBox">
                <form id='auform'>
                    <label>Email
                    <input type="email" name="loginA" />

```

```

        </label>
        <label>Password
            <input type="password" name="password" />
        </label>
    </form>
    <!--Задаємо поля вводу для реєстрації -->

    <a id="logbut" href="#" style="margin-left:
150px;">Авторизація</a>

</div>
<div class="panel__box" id="signUpBox">
    <form id="regform">
        <label>Email
            <input type="email" name='login' />
        </label>
        <label>Password
            <input type="password" name='Password' />
        </label>
        <label>Name
            <input type="email" name='name'>
        </label>
        <label>Second name
            <input type="email" name='secondname'>
        </label>
        <label>MobilePhone
            <input name="number" type="email">
        </label>
        <label>Role
            <select name='role'>

                <option value="student" selected>Студент</option>

                <option value="teacher">Викладач</option>

            </select>
        </label>
        <label>GROUP
            <input type="email" name='group' />
        </label>
    </form>
    <a id='regbut' href="#">Реєстрація</a>

</div>
</div>
</div>
<script src="//static.codepen.io/assets/common/stopExecutionOnTimeout-
de7e2ef6bfe6d24b79a3f68b414b87b8db5b08439cac3f1012092b2290c719cd.js"></script>
>

```


<!--Підключаємо скрипти -->

```
<script>
    var menu = document.getElementById('menu'),
        panelMenu = menu.querySelectorAll('li'),
        panelBoxes = document.querySelectorAll('.panel__box'),
        signUp = document.getElementById('signUp'),
        signIn = document.getElementById('signIn');

    function removeSelection() {
        for (var i = 0, len = panelBoxes.length; i < len; i++) {
            panelBoxes[i].classList.remove('active');
        }
    }

    signIn.onclick = function(e) {
        e.preventDefault();
        removeSelection();
        panelBoxes[0].classList.add('active');
        menu.classList.remove('second-box');
    }

    signUp.onclick = function(e) {
        e.preventDefault();
        removeSelection();
        panelBoxes[1].classList.add('active');
        menu.classList.add('second-box');
    }
    //# sourceMappingURL=pen.js

</script>
```

```
<script src="https://static.codepen.io/assets/editor/live/css_reload-5619dc0905a68b2e6298901de54f73cefe4e079f65a75406858d92924b4938bf.js"></script>
```

```
<script>
    <!--Описуєм спілкування з сервером -->
```

```
$('#regbutform').on('click', function() {
    $('#regbutform').addClass('hide');
    $('#Authorization').addClass('hide');
    $('#Registration').removeClass('hide');
    $('#abut').removeClass('hide');
})
$('#abut').on('click', function() {
    $('#regbutform').removeClass('hide');
```

```

$('#Authorization').removeClass('hide');
$('#Registration').addClass('hide');
$('#abut').addClass('hide');
})
$('#regbut').on('click', function() {

    var fin = 0;
    $('#regform input').each(function(i, elem) {
        if ($(this).val() == '') {
            alert('Заполните все поля');
            fin = 1;
        }
        if($(this).val().length<3)
        {
            alert('Длина поля должна быть больше 3 символов');
            fin = 1;
        }
    });
    if (fin != 1) {
        $.ajax({
            url: 'jq_link.php?a=reg', //url страницы
            (action_ajax_form.php)
            type: "POST", //метод отправки
            dataType: "html", //формат данных
            data: $("#regform").serialize(), // Сериализуем объект
            beforeSend: function() {

            },
            success: function(data) { //данные отправлены успешно
                location.reload();
            },
            error: function(response) { // данные не отправлены
                alert(response);
            }
        });
    }
})
$('#logbut').on('click', function() {

    $.ajax({
        url: 'jq_link.php?a=au', //url страницы
        (action_ajax_form.php)
        type: "POST", //метод отправки
        dataType: "html", //формат данных
        data: $("#auform").serialize(), // Сериализуем объект
        success: function(data) { //данные отправлены успешно\

            location.reload();
        }
    });
}

```

```

        //location.reload();
    },
    error: function(response) { // данные не отправлены
        alert(response);
    }
});
})

</script>
</body>

<!--Опис реалізації чату -->
<body>
<!-- Початок сесії -->
    <div id="modal1" class="modal">
        <div class="modal-content">
            <h3 class="center">Proffile
                <?php echo $_SESSION["name"]." ".$_SESSION["sname"];?>
            </h3>
            <h4>Name:
                <?php echo $_SESSION["name"];?>
            </h4>
            <h4>Second name:
                <?php echo $_SESSION["sname"];?>
            </h4>
            <h4>Mobile number:
                <?php echo $_SESSION["num"];?>
            </h4>
            <h4>Group:</h4>

        </div>
        <div class="modal-footer">
            <a href="#" class="modal-close waves-effect waves-green btn-flat">Закрити</a>
        </div>
    </div>
    <div id="modal2" class="modal">
        <div class="modal-content" id="conProf">

        </div>
        <div class="modal-footer">
            <a href="#" class="modal-close waves-effect waves-green btn-flat">Закрити</a>
        </div>
    </div>

    <div class="container">
        <ul id='dropdown1' class='dropdown-content'>

```

```

        <li><a href="#">one</a></li>
        <li><a href="#">two</a></li>

    </ul>
<!--Модель бокового меню -->

    <div class="row no-padign">
        <div class="col s1 center" id="leftP">

            <a href="example.php" class="center "><i class="medium
material-icons">face</i></a>
            <a href="pretraine.php" class="center"><i class="medium
material-icons">extension</i></a>
            <a href="material.php" class="center act"><i class="medium
material-icons">import_contacts</i></a>
            <a href="examplechat.php" class="center"><i class="medium
material-icons">chat_bubble_outline</i></a>
            <a href="dictionary.php" class="center"><i class="medium
material-icons">book</i></a>
            <a href="test.php" class="center"><i class="medium material-
icons">reorder</i></a>

<!-- Опис форм -->
        </div>
        <div class="col s11" id="rightpan">
            <div class="row">
                <div id='ChatH'>

                    </div>
                </div>
                <div id="send_chat_message">
                    <div class="row">
                        <div id="field_message" style="height: 62vh;
background-color:white;overflow-y: scroll;">

                            </div>
                            <div id="smf" style="height: 10vh">
                                <div class="row">
                                    <div class="input-field col s9 offset-s1">
                                        <i class="material-icons prefix">sms</i>
                                        <input type="text" id="inputMM class="autocomplete">
                                        <label for="input">Отправить сообщение</label>

                                    <div class="input-field col s2">
                                        <a id="sendmessage"><i class="material-icons prefix right">send</i></a>
                                    </div>

                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
<!--Опис поведінки модулів -->

```

```

<script>
<!--Отримання повідомлень-->

        function getMessageChat(chatID) {

                let lastmessagechat =
$('#field_message').children().last().attr('id');
                console.log(lastmessagechat);
                $.ajax({
                        type: 'POST',
                        url: 'jq_link.php?a=getMessageChat',
                        data: 'Chat_id=' + chatID + '&lastmessagechat=' +
lastmessagechat,
                        success: function(data) {

                                $('#field_message').append(data);

                        }
                });
        }
<!--Отримання чату-->
        function getchat() {
                $.ajax({
                        type: 'POST',
                        url: 'jq_link.php?a=getMyChat',
                        success: function(data) {

                                $('#ChatH').append(data);
                                var lid = $('#ChatH').children().last().attr('id');
                                console.log(lid);
                                if (localStorage['lastChannel'] != '') {
                                        lid = localStorage['lastChannel'];
                                }
                        }
                });
        }
<!--Відправка повідомлення -->
        getchat();

        $("#sendmessage").on('click', function() {
                var url = window.location.href;

                var now = new Date();
                var time = now.getFullYear() + '-' + (now.getMonth() + 1) +
                '-' + now.getDate() + ' ' + now.getHours() + ':' + now.getMinutes() + ':' +
                now.getSeconds();

                let message = $("#inputMM").val();
                let counts = (Number($(".MessageC").last().prop('id')) + 1);
                let a = `<div class='MessageC' id='${counts}><div
class='row'>

                <div class='col s6 offset-s1'>
                        <div class='user_m_id' id='?php echo $_SESSION'></div>

```

```

<?php echo $_SESSION['name'].' '.$_SESSION['sname'];?>
    </div>
    <div class = 'col s4 right'>
        <div class='dateM'>${time} </div>
    </div>
</div>
<div class='row'>
    <div class='col s1 offset-s1'>
        <img src='img/10.png' alt='' class='circle responsive-
img'>
    <div class='col s9'>
        <div>
            ${message}
        </div>
    </div>
    `;
    var href = $(location).attr('href');
    var lastIndex = href.lastIndexOf("namechat=");

    href = href.substring((lastIndex + 9), (lastIndex + 18));
    $("#field_message").append(a);
    $.ajax({
        type: 'POST',
        url: 'jq_link.php?a=sendmessage',
        data: 'message=' + message + '&namechat=' + href,
        success: function(data) {

            console.log(data);
        }
    });

})
$('#field_message').on('click', '.user_m_id', function() {

    let id = $(this).attr('id');
    $.ajax({
        type: 'POST',
        url: 'jq_link.php?a=getuserinfo',
        data: 'user_id=' + id,

        success: function(data) {

            $('#conProf').html(data);
            $('#modal2').modal('open');
        }
    });
})

```

ДОДАТОК 3

Веб-додаток для вивчення англійської мови

Опис програми

УКР.НТУУ «КПІ ім. Ігоря Сікорського»_ТЕФ_АПЕПС_ТМ61162_20Б

Аркушів 8

Київ – 2020

АНОТАЦІЯ

Додаток надає можливість користувачеві продуктивно вивчати англійську мову, в будь-якому місці, де є інтернет.

Розроблений веб-додаток дозволяє навчатися шляхом проходження тестів завантажених викладачем, які автоматично перевіряються або проходженням вправ таких як, гра на запам'ятовування слів зі словника, тренування на слух та перевірки власної вимови.

Веб-додаток реалізований з допомогою мов програмування PHP, JavaScript, мовою розмітки HTML та мовою опису зовнішнього вигляду документа CSS, та формальної непроцедурної мови програмування MySQL. Для запуску програмного модуля необхідно встановити OpenServer.

ЗМІСТ

| | |
|--|---|
| 1. Загальні відомості | 4 |
| 2. Функціональне призначення | 5 |
| 3. Опис логічної структури..... | 6 |
| 4. Використовувані технічні засоби | 7 |
| 5. Вхідні і вихідні дані | 8 |

ЗАГАЛЬНІ ВІДОМОСТІ

Відповідно до назви дипломної роботи, розроблений продукт є веб-додатком для вивчення англійської мови, який допомагає покращити процес навчання у вищих навчальних закладах, таких як НТУУ «КПІ».

Користувач для запуску програмного модуля необхідно встановити OpenServer.

Для використання веб-додатку потрібно лише встановити OpenServer та перенести папку з проектом у потрібне місце. Після запуску серверу користувач може взаємодіяти з програмним продуктом.

ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ

Розроблений веб-додаток покликаний вирішити проблему нудного та не продуктивного способу сучасного вивчення англійської мови. Це було реалізовано за допомогою кількох функцій системи, а саме:

- створення розділу, де зберігаються всі матеріали наданні викладачем;
- створення чату, де викладач та студенти можуть взаємодіяти між собою;
- створення особистого словника для кожного користувача;
- створення тестів розроблених викладачем, для студентів;
- надання можливості перегляду статистики користувача.

ОПИС ЛОГІЧНОЇ СТРУКТУРИ

Загальний принцип роботи додатку такий:

- 1) користувач запускає OpenServer;
- 2) у настройках вибирає потрібний йому сайт;
- 3) на початковій сторінці авторизується або реєструється, якщо він є новим користувачем;
- 4) після авторизації користувач переходить на домашню сторінку, де може вибрати потрібний йому розділ;
- 5) натискаючи на потрібний розділ, користувач переходить у цей розділ;
- 6) обравши розділ вправи, користувач може вибрати потрібну йому вправу та пройти її;
- 7) обравши розділ словник, користувач може переглядати потрібні слова;
- 8) обравши розділ тести, користувач може пройти надані йому тести;
- 9) обравши розділ чат, користувач може переписуватись з викладачем або власною групою;
- 10) обравши розділ матеріали, користувач має можливість переглядати надані йому матеріали для вивчення;

Якщо користувач увійшов у систему від імені викладача, тоді обравши розділ тести, у нього відкривається форма для створення тестів, які будуть проходити студенти.

ВИКОРИСТОВУВАНІ ТЕХНІЧНІ ЗАСОБИ

Для запуску програмного модуля проекту необхідно встановити OpenServer та одну з останніх версій його браузера. Додаток розроблений саме для частини цього програмного комплексу — OpenServer.

Користувач не обтяжений встановленням окремих програмних засобів чи інструментів, які потребують спеціальних навичок.

ВХІДНІ І ВИХІДНІ ДАНІ

Вхідними даними є:

- дані користувача;
- матеріали для вивчення;
- слова для словника;
- повідомлення в чаті;
- складові тестів чи вправ для проходження.

Вихідними даними є:

- статистика проходження тестів;
- графічне відображення інтерфейсу користувача.